



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification 7 : <b>G06F 3/033</b></p>	<p><b>A1</b></p>	<p>(11) International Publication Number: <b>WO 00/60440</b></p> <p>(43) International Publication Date: 12 October 2000 (12.10.00)</p>
<p>(21) International Application Number: PCT/US00/07623</p> <p>(22) International Filing Date: 22 March 2000 (22.03.00)</p> <p>(30) Priority Data: 09/287,088 6 April 1999 (06.04.99) US</p> <p>(71) Applicant: VERGICS CORPORATION [US/US]; Suite 210, 467 Saratoga Avenue, San Jose, CA 95129 (US).</p> <p>(72) Inventor: SCHILERU-KEY, Rodica; 945 Daffodil Way, San Jose, CA 95117 (US).</p> <p>(74) Agents: MARINO, Fabio, E. et al.; Skjerven, Morrill, MacPherson, Franklin &amp; Friel LLP, Suite 700, 25 Metro Drive, San Jose, CA 95110 (US).</p>		<p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p><b>Published</b> <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>
<p>(54) Title: GRAPH-BASED VISUAL NAVIGATION THROUGH SPATIAL ENVIRONMENTS</p>		
<p>(57) Abstract</p> <p>A computer system and method of operation thereof are provided that allow interactive navigation and exploration of spatial environments, both real and virtual. The computer system employs a data architecture comprising a network of nodes connected by branches. Each node in the network represents an intersection in the real environment that allows user of the computer system to select which path to follow. In addition to navigating through the environment, the user may also access auxiliary information like video, audio, still images etc. that is related to particular points of interest. The computer system of the present invention is suitable for applications such as developing training materials for complex procedures, building video games or interactive exhibits, authoring stories with multiple alternative endings, creating virtual shopping malls and virtual stores, or allowing virtual visits to remote sites (e.g., museums, historic sites, etc.)</p>		

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

GRAPH-BASED VISUAL NAVIGATION THROUGH SPATIAL  
ENVIRONMENTS

5

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to  
computer programs and, particularly, to computer  
10 systems used to navigate virtual environments.

Related Art

The most common method of presenting a virtual  
tour of a real, physical environment is film or  
videotape. While such a presentation offers high  
15 visual accuracy, the progress of the tour is generally  
limited to a single, static route chosen by the  
photographer or editor. In other words, the observer  
of the tour has none of the choices usually available  
to someone who is physically in the environment:  
20 choices such as turning one way or the other at an  
intersection, skipping uninteresting paths, or  
proceeding directly to a favorite location. The  
observer's interaction with the environment is limited  
to commands such as start, stop, fast forward or  
25 reverse. A further limitation of this approach is that  
the observer generally has no other context than the  
image, itself. That is, the limited field of view  
makes it difficult for the observer to orient  
him/herself in a complex environment.

30 For tours of virtual environments, computer  
systems are commonly used to render images from models  
representing the virtual environments. As opposed to  
videotaped presentations, such systems can be highly  
interactive since the images are rendered in real time.

There is, however, a tradeoff between cost and visual accuracy. Computer systems capable of rendering very high-resolution images in real time are prohibitively expensive, while more affordable systems produce  
5 inferior quality images.

#### SUMMARY OF THE INVENTION

The present invention relates to a computer system which allows interactive navigation and exploration of  
10 spatial environments, both real and virtual. The computer system employs a data architecture comprising a network of nodes connected by branches. Each node in the network represents an intersection in the real environment that allows a user of the computer system  
15 to select which path to follow. Likewise, each branch in the network represents a path connecting physical intersections in the real environment.

The network is constructed directly from a map of the target environment. Navigation data such as image  
20 frame sequences, intersections between paths, and other related information are associated with the elements of the network. This establishes a direct relationship between locations in the environment and the data which represent them. From such an organization, the user  
25 may tour the environment, viewing the image sequences associated with each path and choosing among intersecting paths at will.

In addition to navigating through the environment, the user may also access auxiliary information that is  
30 related to particular points of interest. This auxiliary information can be of varying forms: video, audio, still images, etc. Such information is linked to a point of interest through the associated network element. By associating this information with the

virtual environment, the computer system of the present invention is able to provide the observer with an enhanced view of the real environment represented by the virtual environment.

5

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that illustrates the architecture of the principal design constructs of the present invention;

10        Fig. 2 illustrates an example of the network of nodes and branches that is the underlying representation of the navigation model;

Fig. 3 illustrates one possible map that can be realized from the network shown in Fig. 2;

15        Fig. 4 is a functional flow diagram that illustrates the logic and actions of the image frame sequence processing function;

Figs. 5a, 5b and 5c illustrate the rotation of path vectors from absolute to user-relative  
20        coordinates;

Fig. 6 is a functional flow diagram that illustrates the process of initializing a new branch for playback.

25        Fig. 7 illustrates the calculation of branches and image frames for the Browse event.

Fig. 8 illustrate the hardware and software structure of a computer system, in accordance to an embodiment of the invention.

30        Fig. 9 illustrates the derivation of the software classes that comprise the application library.

Fig. 10 illustrates the derivation of authoring classes from the application library.

Figs. 11-16 show a window displayed on a screen of a display device of the computer system of Fig. 8, at

successive points during navigation of a virtual environment.

Figs. 17-22 show a window displayed on a screen of a display device of the computer system of Fig. 8, at successive points during authoring of a virtual environment.

#### DETAILED DESCRIPTION OF THE INVENTION

As disclosed herein, the present invention includes a method and a computer software system for composing and executing interactive tours of spatial environments. Fig. 8 illustrates the hardware and software context of the present invention. The computer hardware specified is representative of modern multimedia computer systems. Such systems contain a central processing unit (CPU) and memory, a video card to generate images, and a sound card to generate sound. The images are displayed on an electronic computer display monitor. One example of a suitable CPU and memory for the computer system is a Pentium processor, 200 Mhz, with 32 MB RAM. The display must be capable of 16-bit color.

The computer hardware further includes a keyboard and pointing device for data input. As is understood by those skilled in the art, the pointing device may be a mouse, trackball, joystick, or other similar device. The computer system further includes a mass storage device and/or a network interface. Either a mass storage device or a computer network may be used to store large data such as image files. Suitable mass storage devices will have the characteristics of very large capacity as well as very fast, random access. An example of a suitable mass storage device is a CD-ROM drive. An example of a suitable computer network is

the World Wide Web.

The software context described in Fig. 8 includes a computer operating system and several service layers of computer software. Examples of suitable operating systems are Windows95, Windows98 and Windows NT 4.0, available from Microsoft Corp. of Redmond, Wash. The present invention requires services provided by three software service applications, NetShow, DirectShow, and DirectX/Direct3D, also available from Microsoft Corp.

10 As illustrated in Fig. 8, the application software of the present invention has interfaces with each of these three service applications.

Those skilled in the art realize that other hardware/software platforms can be used in lieu of the ones described herein in accordance to the principles of the present invention. Accordingly, the invention is not limited to any particular hardware/software platform.

Fig. 1 illustrates the architecture of the present invention. Rectangles in the figure identify design constructs, while arrows between rectangles identify relationships between constructs. As illustrated by the legend in the lower right end corner of Fig. 1, arrows with a single head on each end (e.g., arrow 121) indicate a one to one relationship between corresponding constructs, while arrows with a single head on one hand and a double head on the other end (e.g., arrow 123) indicate a one-to-many relationship between corresponding constructs.

Documents 101 are the primary construct of the architecture, representing a complete navigational model. Views 102 display some aspect of the document in a window of the computer display. Each view 102 is typically associated with a display window.

Relationship 123 indicates that a single document 101 may be displayed using multiple views 102.

Relationship 125 indicates that three distinct types of views are used: main views 105, map views 106, and  
5 popup views 107. In one embodiment, main view 105 displays image frame sequences associated with each path on the tour, as illustrated by arrow 128. Main view 105 also contains user controls 109, as illustrated by arrow 129. Map view 106, in turn,  
10 displays an interactive map of the environment, including tour paths, intersections, and points of interest (HotSpots) that allow users of the computer system to access auxiliary information. Popup views 107 display auxiliary data 110 associated with  
15 HotSpots, as illustrated by arrow 130.

Each document 101 may manage multiple streams 103, as illustrated by arrow 122. Each stream 103, in turn, provides data to a view 102, as illustrated by arrow 121. These data is contained in one or more stream  
20 files, whose names are stored in model database 104. Each document 101 is associated with a single model database 104 that contains the complete data for the document's navigation model, as illustrated by arrow 124. Model database 104 contains multiple model  
25 elements 108 that make up the navigation model, as illustrated by arrow 126. Relationship 132 indicates that four kinds of model elements are used. Nodes 118 represent intersections. Pairs of nodes 118 are connected by branches 119, that represent paths between  
30 intersections, as illustrated by arrow 135. A single node 118 may serve as a connecting point for two or more branches 119. HotSpots 120 represent points of interest that enable users to access auxiliary data 110. HotSpots are placed along branches 119, as



illustrated by arrow 137. Branch links 117 allow users to transition from one model document to another, as illustrated by arrow 131. Branch links 117 allow traversal between independent navigation models, represented by different documents. Each model element 108 is displayed in a map view 106 as an element of the interactive map, as illustrated by arrow 127.

Relationship 133 indicates that auxiliary data 110 may take several forms. Model links 111 are similar to branch links 117 insofar as they connect documents 101, as illustrated by arrow 139. However, while a branch link jumps unconditionally to a new model, replacing the current model, a model link initiates a new model within the context of the current model. When a new model is terminated, program execution returns to the location of the model link. A web link 112 connects the model to a World Wide Web page. When a web link 112 is executed, the program launches a user's default web browser to access a URL associated with web link 112. In such case, the browser acts as a popup view 107. Auxiliary data 110 displayed in a popup view 107 may also include single images 113, image frame sequences 114, audio clips 115, and text documents 116. When display of auxiliary data 110 is terminated, execution returns to the location of HotSpot 120 in the original model. Users may terminate display of auxiliary data 110 either by closing popup view 107 or by selecting one of the controls in main view 105.

Fig. 2 illustrates a graphical representation of the navigational model, hereafter referred to as "the model". The model is conceptually a network of nodes and connecting branches. The network is represented as a directed, cyclic graph. Each node of the graph represents an endpoint or intersection in the model,

and each branch represents a path in the model. For example, branch 207 connecting nodes 202 and 203, represents a path in the model between the intersections represented by nodes 202 and 203.

- 5 Furthermore, the orientation of the arrow on branch 207 indicates that the associated path is directional, leading from node 202 to node 203.

In one embodiment, the model of Fig. 2 is displayed to the user as a map of the environment, as  
10 shown in Fig. 3. Thus, intersections 301-306 of Fig. 3 correspond to nodes 201-206 of Fig. 2. Paths on the map are not necessarily straight lines (i.e., a path may be composed of multiple, connected line segments). Path 307, constructed of two segments, corresponds to  
15 branch 207. In addition to intersections and paths, the map also displays HotSpots located along paths. For example, HotSpot 309 is located along path 308. Since in the model HotSpots are associated with branches, HotSpot 309 is associated with branch 208  
20 that, in turn, corresponds to path 308.

Each branch in the model is associated with an image sequence that is displayed to the user during the course of the tour. In one embodiment, the sequence is a video clip shot while moving along the associated  
25 path in the real environment. The sequence may take other forms as well, such as computer-animated image frames showing a path through a virtual environment. Image sequences may reside on the host computer, or may reside on remote servers located anywhere on the World  
30 Wide Web. For instance, in some embodiments, image sequences may be shown using streaming video.

HotSpots represent a linkage between an area of a navigation frame, typically an object in the frame, and a set of associated data. A HotSpot is represented on

the map by an icon and in the main view image display by a bounded region of interest (ROI). A HotSpot's associated data may take a variety of forms including, but not limited to, video sequences, animations, audio clips, text, still images, and World Wide Web links. When a HotSpot is activated during the course of the tour, the associated data is displayed in a popup view window.

Given the underlying structure of the data, the following sections describe the operations that manipulate the data. The present invention provides three classes of capabilities: initialization, authoring (constructing and modifying the model), and playback (navigating the model).

#### 15 Overview of Program Execution

When the program is started, a map view window 1120 (Fig. 11) is initialized with intersections, paths, and HotSpots retrieved from model database 104. The default starting intersection, as defined in model database 104, is the initial start position. Main view window 1110 shows the image frame associated with the starting intersection, as well as any path choices 1130 leading away from the starting intersection.

When the user selects a choice, the path represented by that choice and any destination intersections are highlighted in map view window 1120 (e.g., path 1210 and node 1220 in Fig. 12). The program plays the image sequence associated with the path (Fig. 12). Any HotSpots on the current path are highlighted on the map as the image sequence nears their locations, and the user receives both a visual and audio queue to the HotSpot's proximity (Figs. 13-16).

As each image frame is decoded, the program updates the current frame number (or sample time).

When the current frame approaches the end of the sequence range associated with the current path, the program displays the path choices available at the upcoming intersection (Fig. 11). If the user does not  
5 select a choice before the image sequence reaches the intersection, image playback is paused. If the user does select a choice, image playback proceeds directly with the image sequence for the selected path (i.e. image playback is not paused at the intersection).

10 The user may also control program execution in other ways. For example., the user can pause and continue image playback at will, and can jump to an arbitrary intersection by simply selecting a corresponding icon in map view window 1120. Similarly,  
15 the user can jump to any HotSpot in the model by selecting a corresponding icon in map view window 1120.

#### Initialization

In the case of the authoring tool, the program is initialized by simply starting the software  
20 application. In the case of the player the following operations are performed:

#### Startup

This operation first creates a document, a model database, a stream object and a main view. A reference  
25 to the model database is then passed to the stream object and a reference to the stream object is added to the document.

#### Load Model File

This operation opens a model database file,  
30 connects the main view to the stream object created during Startup, creates a map view, and connects the map view to the main view stream object. The model database created during Startup is then populated with the contents of the model database file. A map is

drawn using a bitmap file. Path, intersections, and HotSpots from the model are overlaid on the background. The start position is initially set to the start node and the intersection icon is highlighted on a map  
5 displayed in map view window 1120.

#### Display Initial Image

This initial stream is opened and the image frame associated with the starting node is displayed in main view window 1110. User controls are attached to main  
10 view window 1110 and the stream is set to Pause, waiting for user input.

#### Authoring

The general process of authoring is to construct a model by drawing elements in a map view window 1720  
15 (Fig. 17) that are viewed in main view window 1710. Construction of a model typically begins with a diagram of the environment, a set of image sequences, and auxiliary data. The diagram is displayed as a bitmap in map view window 1720. The user defines model  
20 elements by drawing them over the diagram. For each path defined, the user selects an image sequence from main view window 1710 and attaches the image sequence to the path. For each HotSpot defined, the user selects auxiliary data and associates the auxiliary  
25 date with the HotSpot. Editing operations allow the user to modify the contents of any model element. The authoring program enables the user to perform the following operations:

Add Node: The user designates the location of the  
30 node on the map displayed in map view window 1720. This operation draws an intersection icon at the designated location, as shown in Fig. 17, creates the node in the model, and records the location of the intersection in the node.

Delete Node: The user selects an intersection icon on the map displayed in map view 1720. This operation removes the intersection icon from the map and deletes the associated node from the model. If the node is the  
5 start or end point of any branches, the branches are deleted as well (see operation Delete Branch).

Edit Node: The user selects an intersection icon on the map displayed in map view window 1720. This operation then displays a dialog 1910 (Fig. 19) wherein  
10 the user can modify any of the data contained in the associated node 1920.

Add Branch: The user draws a path on the map displayed in map view window 1720. To draw the path, the user starts at any intersection and then draws a  
15 series of connected line segments 1810 that end at another intersection (Fig. 18). The shape of the path should closely approximate the shape of the actual path followed to produce the image sequence for the path. The user then designates a start and an end of the  
20 image sequence for the branch by selecting image frames in main window 1710. This operation creates a branch in the model, records a specification of the path (including all segments) in the branch, records references to the start and end nodes in the branch,  
25 and adds a reference to the branch to its start and end nodes.

Delete Branch: The user selects a path on the map displayed in map view window 1720. This operation removes the path from the map displayed in map view  
30 window 1720, deletes the associated branch from the model, and removes references to the branch from the two nodes connected by the branch.

Edit Branch: The user first selects a path on the map displayed in map view window 1720. Then a dialog

2010 is displayed to allow the user to modify any of the data contained in the associated branch 2020, as shown in Fig. 20.

Add HotSpot: The user defines a HotSpot in main window 1710 with the image sequence paused at the desired frame (the image sequence must already have been assigned to a branch). The user then defines a HotSpot area by drawing a bounding figure around a region of interest 2120 in main view window 1710 (Fig. 21). The user specifies data to be associated with the HotSpot in a popup dialog 2110 (Fig. 21). This operation uses the offset of the image frame within the sequence to interpolate a position along the associated path on the map. Based on this information a Hotspot icon 2220 is drawn at a corresponding location on the map (Fig. 22).

Delete HotSpot: The user selects a HotSpot icon on the map displayed in map view window 1720. The HotSpot icon is removed from the map, the associated HotSpot is deleted from the model and removes any references to the HotSpot are deleted from all branches associated with the HotSpot.

Edit HotSpot: The user selects a HotSpot icon on the map displayed in map view window 1720. A dialog box is then displayed to allow the user to modify any of the data contained on the corresponding HotSpot.

Add Branch Link: The user draws a path on the map displayed in map view window 1720, and optionally associates the path with an image sequence, in the manner specified with respect to the Add Branch operation. The user then specifies both a model document in database 104 and a node within that document's model as the endpoint of the branch link. This operation creates a branch link in the model,

records a specification of the path (including all segments) in the branch link, records references to start and end nodes in the branch link, and adds a reference to the branch link to the start and end  
5 nodes. Note that the end node is not reached by traversing the branch link; the destination is the remote node.

Delete Branch Link: The user selects a path on the map displayed in map view window 1720. The path is  
10 removed from the map, the associated branch link is removed from the model, and only references to the branch link are removed from the nodes connected by the branch link.

Edit Branch Link: The user selects a path on the  
15 map displayed in the map view 1720. A dialog box is then displayed to allow the user to modify any of the data contained in the associated branch link.

Save Model: The user is prompted for a file name, and the contents of the Model Database are written to  
20 the file specified by the user

#### Playback

The playback capability allows a user to execute the model, resulting in an interactive tour of the virtual environment. There are two separate functions  
25 involved in playback: processing image frames and handling user input. Conceptually, the processing of image frames occurs iteratively in the main processing loop of the program, while user input is handled asynchronously.

30 The program initiates the playback capability by preparing map view window 1120. First a static background image is drawn. This is typically a building floor plan or other representation of the physical environment. Next the program draws each



path, intersection, and HotSpot represented in model database 104. The program displays an image frame associated with the default initial node in main view window 1110. The program further displays available choices of outgoing branches from the initial node. Once the user selects a branch to follow, playback begins. The program highlights the selected path and the path's destination intersection on the map and begins real-time display of the image frame sequence.

10 As execution continues along a branch, the user may pause and execute any HotSpot encountered along the path. As execution reaches the end of the path, the program displays available branch choices for the upcoming node. The user makes a selection to continue

15 execution.

Fig. 4 is a flowchart of image frame processing operation 400. Operation 400 first looks for certain conditions that require special processing. When all conditions have been handled, the next available image frame is rendered. First, stage 405 determines whether the active stream is set to a Run state. If the stream state is not set to Run, image playback is paused and operation 400 terminates. Next, stage 410 determines whether the current frame is close to a HotSpot.

20 Proximity is determined either with respect to a number of frames or to a time delta. If the test is true, in stage 415 the HotSpot is activated (e.g., by highlighting the HotSpot icon in map view window 1120 and alerting the user to the proximity of the HotSpot

25 via a visual and/or auditory queue).

30 Stage 420 then determines whether the current frame is at the end of the current branch. Proximity is again determined either with respect to a number of frames or to a time delta. If the test is true, path

choices are activated in stage 425. Path choices are the set of paths branching out from the upcoming intersection. Activation includes first calculating the set of path choices and then displaying the choices  
5 for user selection. Calculation is required because path choices are always displayed relative to the current direction of travel (i.e. from the user's viewpoint), rather than according to a fixed coordinate systems. Fig. 5a illustrates a sample scenario where  
10 path 501 enters intersection 502, and paths 503 and 504 leave intersection 502. If the current playback location is along path 501, the choices presented to the user at intersection 502 are to take either left path 503 or right path 504. Fig. 5b illustrates these  
15 choices from the user's viewpoint. Since display of path choices is from the user's viewpoint, the user's current direction of travel corresponds to straight upscreen, 90 degrees left and right correspond to straight left and right, respectively, on the screen,  
20 while 180 degrees backwards corresponds to straight downscreen.

Calculation of path choices is accomplished as follows. Recalling that a branch may represent a path of multiple line segments, the geometrical orientation  
25 of an outgoing branch is determined by a vector of a first line segment of the branch. The orientation of an incoming branch is likewise determined by a vector of a last segment in the branch. The incoming direction vector (representing the user's point of  
30 view) is rotated such that the vector points "up" (i.e., the vector aligns with a 90 degrees angle). The difference angle between the original orientation of the incoming vector and the rotated vector is the angle by which all of the outgoing branch's vectors are

rotated. In Fig. 5c, segment BA is the vector of the incoming branch (path 501), and segments AC and AD are the vectors of outgoing branches 503 and 504. For each outgoing directional vector (xy),

$$\begin{aligned} \text{newX} &= x * \cos \theta - y * \sin \theta \\ \text{newY} &= x * \sin \theta + y * \cos \theta \end{aligned}$$

where  $\theta$  is the difference angle.

As shown in Fig. 5c,  $\sin \theta$  and  $\cos \theta$  represent components  $V_x$  and  $V_y$ , respectively, of the incoming vector, so there is no need to compute the actual difference angle.

Finally, stage 430 determines whether the current frame is the last frame of the branch. If the test is true, stage 435 determines whether the user has chosen a branch to proceed on. Otherwise the stream state is set to Pause in stage 445 and operation 400 terminates. If a branch choice is available, the image frame sequence that corresponds to the chosen branch is initialized in stage 440, as illustrated in Fig. 6 (operation 600). First, the location of the next image frame sequence is retrieved from the new branch in stage 610. Stage 615 then determines whether the next sequence is stored in a different file from the current sequence, in which case the stream of the old file is closed in stage 620 and a stream for the new file is opened in stage 625. The program then seeks to the first frame of the sequence, in stage 630. The path on the map that corresponds to the new branch is highlighted in stage 635. Finally, the program sets the stream state to Run in stage 640 and operation 600 terminates.

The final step in operation 400 is to render the next image frame in main view window 1110 in stage 450.

During image frame processing, the user can manipulate various controls to control playback. The present invention interleaves handling of asynchronous input events with image frame processing. The following sections describe the processing of possible input events.

Choose Branch: Each node in the model has one or more branches leading away from the node. In two instances, when the tour reaches a node and when an input event causes a jump to a node, the user must choose a branch to follow away from the node. The program displays branch choices available to the user as control icons in main view window 1110. The user selects one of the icons, indicating the preferred choice (Fig. 12). This action triggers the Choose Branch input event. The program processes the event by storing the chosen branch. The choice is then retrieved and used by the image frame processing loop as described above.

Jump to Node: At any point, the user may cause the image frame display to jump to an arbitrary node in the model by selecting an associated intersection in map view window 1120. This action triggers a Jump to Node input event. Upon receiving the event, the program identifies the image frame associated with the selected node. If the new image frame is stored in a different file than the previous image frame sequence, the program closes the previous input stream and opens a stream to the new image file. The program then sets the stream state to Pause and displays the image frame associated with the new node in main view window 1110. Note that this image frame is the final frame of one of the incoming branches. The program highlights the associated path in map view window 1120 to help orient

the user at the new location. The program then calculates branch choices for the node and displays them in a correct orientation relative to the user's view point (Fig. 11).

5        Jump to HotSpot: At any point, the user may cause the image frame display to jump to an arbitrary HotSpot in the model by selecting an associated HotSpot icon (e.g., HotSpot icons 1320 and 1520 in Figs. 13 and 15, respectively) in map view window 1120. This action  
10 triggers a Jump to HotSpot input event. The processing of this event is similar to the processing of a Jump to Node event. The program initializes the image frame sequence for the branch on which the HotSpot is located. The program then displays a particular image  
15 frame associated with the HotSpot and highlights the new path on the map. Finally, the program overlays the HotSpot's bounded region of interest on the displayed video frame and highlights the HotSpot icon on the map (Figs. 13 and 15).

20        Open HotSpot: Whenever a HotSpot region of interest is displayed in main view window 1110, the user may open the corresponding HotSpot by selecting the region of interest. This action triggers an Execute HotSpot event. Upon receiving this event, the  
25 program first determines the type of document (e.g. text, image, audio) referred by the HotSpot. The program then pops up an appropriate type of window to display the data contained in the document (e.g., popup windows 1410 and 1610 in Figs. 14 and 16,  
30 respectively). The program then opens the document and displays the document's contents in the pop up window. For streaming types of data (i.e., audio clips, video clips or image sequences), the program creates the stream and initializes a popup window to the starting

frame specified by the HotSpot. User controls are then attached to the popup window and the stream state is set to Run. If the HotSpot is an interactive game or exhibit a database model file is loaded, but no map  
5 view window is created, since the map represents a conceptual flow.

Pause: At any point during playback, the user may cause the image frame display to stop at the current frame by selecting a Pause control 1150 in main view  
10 window 1110. Upon receiving this event, the program sets the stream state to Pause, thereby preventing the image display function from displaying subsequent frames.

Play: Once playback is performed, the user may  
15 cause image frame display to resume from the paused state by selecting Play control 1140 in main view window 1110. Upon receiving this event, the program sets the stream state to Run, thereby enabling the image display function to display subsequent frames.

20 Fast Forward: During image frame playback, the user may cause the program to increase the speed of frame display by selecting Fast Forward control 1170 in main view window 1110. Upon receiving this event, the program displays the remainder of the current image  
25 frame sequence (i.e., the current path) at double the normal speed rate (some frames may be skipped during this operation).

Fast Reverse: During image frame playback, the user may cause the program to display image frames in  
30 reverse order by selecting Fast Reverse control 1160 in main view window 1110. Upon receiving this event, the program displays the current image frame sequence in reverse order at double the normal speed rate (some frames may be skipped during this operation).

Browse: At any point, the user may cause the program to scan through all of the image frames in the model by use of the Browse control 1180 in main view window 1110. Browse control 1180 is continuously  
5 variable over a range of values. When Browse control 1180 is activated an event is opened that contains two values: a state value that indicates whether the control is engaged or disengaged and an index value that indicates the position of the control. Upon  
10 receiving this event, the program sets the stream state to Pause. If the control state is Engaged, the program maps the index value to one of the branches in the model and highlights the associated path on the map. If the control state is Disengaged, the program maps  
15 the index value and highlights a path as described above. The program further calculates a particular image frame within the branch's frame segment, retrieves that frame, and displays the frame in main view window 1110.

20 The image frame calculation is based on an arbitrary serial ordering of all branches in the model. This function maps the range of Browse index values over the resulting serial sequence. A specific index value is then mapped to an offset within a specific  
25 branch, the image frame at that offset in the branch's associated image frame sequence is then identified. Fig. 7 illustrates this function. Paths 705, 706, and 707 represent the branches of a simple model. They are associated with frame sequences 702, 703, and 704,  
30 respectively. A range of index values 701 maps to the three sequences, as shown in Fig. 7. For example, an index value of 60 falls within the subrange that maps to sequence 703, and the program highlights path 706 accordingly. Once the offset of the given index value

within the subrange is calculated the index value is further, it is straightforward to mapping to frame 5 in frame sequence 703.

Figs. 9-10 show the architecture of the application software classes that implement the present invention. As illustrated in Fig. 9, the application software comprises a set of classes derived from the Microsoft Foundation Classes (MFC), as well as application classes that make up a separate application library. In addition to inheritance relationships between classes, the application specific classes are related to one another through pointer references. MFC-derived classes (VITAppObj, CMultiDocTemplate, CNetPlayerDoc, CNetPlayerView, CNodesView, CChildFrame, and CPlayCtrls) provide the operations that are specific to Microsoft Windows user interface. Examples of such operations are creating and deleting windows and receiving events from the pointing device.

Application specific library classes implement the core functionality of the program of the present invention. For instance, these classes implement the playback capability. The role of each of the application library classes is as described below:

CDocHelper: This class creates and stores each of the stream objects (CMMStream) needed by the document. This class also, keeps track of the currently active stream.

CNodesNetDB: This class stores and provides access to the elements of the model.

CStreamViewHelper: This class performs processing required for stream views (the main view and streaming popup views).. This class further interprets and processes events (e.g. user controls and HotSpot invocation) received from one of these views by class



CnetPlayerView.

CMMStream: This class performs processing required for streams (e.g. opening and closing stream files).

5 CNodesViewHelper: This class performs processing required for the map view. This class further interprets and processes events (e.g., jumping to intersections) received from the map view by class CnodesView.

10 CNodesNetGraph: This class draws model elements in map view window 1120.

A set of application classes derived from the application library classes add editing operations to the basic library operations. For example, these classes implement the authoring capability. Fig. 10  
15 illustrates the relationships between library classes and authoring classes. It will be apparent to those skilled in the art that the operations of the playback program are a subset of those of the authoring program. That is, the authoring program contains every operation  
20 of the playback program, plus additional operations. These classes, and the additional operations that they provide, are described below:

CDocHelperA: This class provides the capability to select a bitmap file (static background) for the  
25 document.

CNodesNetDBA: This class modifies the model database, both adding and deleting model element data.

CStreamViewHelperA: This class provides operations to modify stream views, specifically adding HotSpots to  
30 the view. This class invokes operations in CNodesNetDBA to update the database for new HotSpots.

CNodesViewHelperA: This class provides operations to modify the map view, specifically adding, deleting, and editing the model elements that are represented in

map view window 1120 (e.g., nodes, branches, branch links). This class further invokes operations in CNodeNetGraphA to update the drawings of model elements in the map view.

- 5        CNodeNetGraphA: This class provides operations to modify the drawings of model elements in the map view.

A further embodiment of the present invention uses the model as the underlying representation of a video game. A video interactive game can be scripted entirely  
10 without programming or geometric modeling by constructing a graph of decision nodes with branches. The outgoing branches of a node represent actions or decisions that are taken in the flow of the game. Each outgoing branch has a distinct orientation vector. The  
15 game associates each unique vector with a user interface control. For example, assume a simple control device with three buttons (e.g. a three-button mouse). All vectors in the graph that are oriented at 90 degrees could be controlled by interface button A, all  
20 vectors at 45 degrees by button B, and all vectors at 180 degrees by button C.

Assume the game is initialized to start at some default node in the graph. If from that location the user pushes button A, the game follows the branch  
25 oriented at 90 degrees. This brings the user to the node at the end of this branch. If at this node the user presses button B, the branch at 45 degrees is followed, thus connecting to the next node. If there is no branch that corresponds to the input, the game  
30 position is not advanced. The game may also have disconnected nodes (nodes that cannot be reached by a branch in the graph) or, in fact, multiple disjunct graphs. In this case, the user interface must provide the capability to pass node identifiers to the program

in order to effect jumps to disjunct nodes. Note that in this embodiment, the graph is not displayed as a map since it functions as a decision graph, rather than as a map of a spatial environment.

5       It will be apparent to those skilled in the art that the same program code is used to traverse the graphs in both embodiments. The difference between the environments resides in the format of the user interface.

10       Furthermore, application can be extended to support multiple players. Players navigating the same model can be represented on the map with different symbols and placed on appropriate locations on the branches they travel on.

15       Application can be integrated with other collaborative programs such as chat (text or voice), (e.g., Netmeeting). Players can then open a communication channel with other players by simply clicking on symbols corresponding to the other players.

20       Furthermore, computer-generated graphics can be integrated with the video to allow for more complex animations composited within video to support interactive 3-D games.

      The present invention can also be applied to  
25   training materials for complex procedures. For example, a video illustrating the actual procedure taking place can be shown next to a conceptual flow of the presentation shown in map view window 1120 with the branches highlighting the current stage in the  
30   procedure. Additional types of hot spots can be inserted by user such as personal notes and links to other pertinent information to help the learning process.

      In addition, video games or interactive exhibits

can be scripted by building the network of decision points and branches, with no special programming (already demonstrated in the current embodiment).

Furthermore, stories with multiple alternative  
5 endings can be developed in accordance to the principles of the invention. For instance, an application can be built where kids assemble segments of the story in different ways to create new stories.

Virtual shopping mall and virtual stores can also  
10 be developed in accordance to the principles of the invention. Currently on-line shoppers must know in advance exactly what they want to search within a category. By associating Hot spots to items shown in a moving video of a real store, the computer system of  
15 the present invention replicates a real life shopping experience, allowing, for example, impulse buying. Users can obtain more information about an item shown in the video by simply clicking on a corresponding hot spot. The item is, in turn, associated with unique URL  
20 of the site's e-commerce database. Users can thus add the item to their electronic shopping basket.

Finally, the computer system of the present invention is also suitable for high-bandwidth applications. Video files encoded according to a  
25 broadcast quality standard such as MPEG2 can be streamed over high-speed networks to allow virtual visits to remote sites (e.g., museums, historic sites, etc).

Embodiments described above illustrate but do not  
30 limit the invention. In particular, the invention is not limited by any particular graphical user interface. In fact, any graphical user interface known in the art to perform the functions described herein can be used in accordance to the principles of the invention. In

addition, the invention is not limited to any particular hardware/software implementation. Those skilled in the art realize that alternative hardware software implementations can be employed in lieu of the one described herein in accordance to the principles of the present invention. Other embodiments and variations are within the scope of the invention, as defined by the following claims.

CLAIMS

I claim:

1. A computer system for navigating a virtual environment, the computer system comprising:
  - 5 a computer;
  - a display device connected to the computer;
  - one or more input devices connected to the computer; and
  - a computer program executing on the computer,
- 10 wherein the computer program comprises computer instructions for:
  - generating a virtual environment from a real environment, the virtual environment comprising a plurality of nodes connected by branches, wherein
  - 15 each node represents an intersection in the real environment and each branch represents a path connecting intersections in the real environment;
  - displaying a map of at least a portion of the nodes and branches of the virtual environment in a
  - 20 first window of the display device;
  - displaying a visual representation of the virtual environment in a second window of the display device; and
  - enabling a user to navigate through the
  - 25 virtual environment by entering navigation commands via at least one input device.
2. The computer system of claim 1, wherein each path in the map further comprises directional
- 30 information, and the directional information is expressed relative to a current position in the map.
3. The computer system of claim 1, wherein the computer program further comprises computer

instructions for:

associating multimedia content with a portion  
of the virtual environment;

5 displaying a visual representation of the  
multimedia content as an element of the virtual  
environment displayed in the second window of the  
display device;

detecting that the user has selected the  
visual representation of the multimedia content  
10 via one of the input devices; and

displaying the multimedia content in a window  
of the display device.

4. The computer system of claim 3, wherein the  
15 multimedia content is displayed in a third window of  
the display device.

5. The computer system of claim 3, wherein the  
multimedia content further comprises a web page.  
20

6. The computer system of claim 3, wherein the  
multimedia content further comprises a video clip.

7. A computer system for composing a virtual  
25 environment, the computer system comprising:

a computer;

a display device connected to the computer;

one or more input devices connected to the  
computer; and

30 a computer program executing on the computer,  
wherein the computer program comprises computer  
instructions for:

generating a virtual environment from a real  
environment, the virtual environment comprising a

plurality of nodes connected by branches;

enabling a user to associate an intersection of the real environment with a node of the virtual environment; and

5           enabling a user to associate a visual representation of a path connecting intersections of the real environment with a branch of the virtual environment.

10           8. The computer system of claim 7, wherein the computer program further comprises computer instructions for enabling a user to associate multimedia content with an element of the virtual environment.

15           9. The computer system of claim 7, wherein the computer program further comprises computer instructions for editing associations between nodes and intersections and between branches and paths.

20           10. A method for navigating a virtual environment generated from a real environment, the virtual environment comprising a plurality of nodes connected by branches, wherein each node represents an  
25 intersection in the real environment and each branch represents a path connecting intersections in the real environment, the method comprising:

displaying a map of at least a portion of the nodes and branches of the virtual environment in a  
30 first window of a display device;

displaying a visual representation of the virtual environment in a second window of the display device; and

a user navigating through the virtual



environment by entering navigation commands via at least one input device connected to the display device.

5        11. The method of claim 10, wherein each path in the map further comprises directional information, and the directional information is expressed relative to a current position in the map.

10        12. The method of claim 10, further comprising:  
         associating multimedia content with a portion  
         of the virtual environment;  
         displaying a visual representation of the  
         multimedia content as an element of the virtual  
15        environment displayed in the second window of the  
         display device;  
         the user selecting the visual representation  
         of the multimedia content via one of the input  
         devices; and  
20        displaying the multimedia content in a window  
         of the display device.

         13. The method of claim 12, wherein the  
         multimedia content is displayed in a third window of  
25        the display device.

         14. The method of claim 12, wherein the  
         multimedia content further comprises a web page.

30        15. The method of claim 12, wherein the  
         multimedia content further comprises a video clip.

         16. A method for composing a virtual environment,  
         the method comprising:

generating a virtual environment from a real environment, the virtual environment comprising a plurality of nodes connected by branches;

5 a user associating an intersection of the real environment with a node of the virtual environment; and

10 the user associating a visual representation of a path connecting intersections of the real environment with a branch of the virtual environment.

17. The method of claim 16, further comprising the user associating multimedia content with an element of the virtual environment.

15

18. The method of claim 16, further comprising editing associations between nodes and intersections and between branches and paths.

20 19. A computer-readable storage medium comprising computer instructions for:

25 generating a virtual environment from a real environment, the virtual environment comprising a plurality of nodes connected by branches, wherein each node represents an intersection in the real environment and each branch represents a path connecting intersections in the real environment;

30 displaying a map of at least a portion of the nodes and branches of the virtual environment in a first window of a display device;

displaying a visual representation of the virtual environment in a second window of the display device; and

enabling a user to navigate through the

virtual environment by entering navigation commands via at least one input device.

20. The computer-readable storage medium of claim 5 19, wherein each path in the map further comprises directional information, and the directional information is expressed relative to a current position in the map.

10 21. The computer-readable storage medium of claim 19, further comprising computer instructions for:  
associating multimedia content with a portion of the virtual environment;  
displaying a visual representation of the  
15 multimedia content as an element of the virtual environment displayed in the second window of the display device;  
detecting that the user has selected the visual representation of the multimedia content  
20 via one of the input devices; and  
displaying the multimedia content in a window of the display device.

22. The computer-readable storage medium of claim 25 21, wherein the multimedia content is displayed in a third window of the display device.

23. The computer-readable storage medium of claim 21, wherein the multimedia content further comprises a  
30 web page.

24. The computer-readable storage medium of claim 21, wherein the multimedia content further comprises a video clip.

25. A computer-readable storage medium comprising computer instructions for:

- generating a virtual environment from a real environment, the virtual environment comprising a plurality of nodes connected by branches;
- enabling a user to associate an intersection of the real environment with a node of the virtual environment; and
- enabling the user to associate a visual representation of a path connecting intersections of the real environment with a branch of the virtual environment.

26. The computer-readable storage medium of claim 25, further comprising computer instructions for enabling the user to associate multimedia content with an element of the virtual environment.

27. The computer-readable storage medium of claim 25, further comprising computer instructions for enabling the user to edit associations between nodes and intersections and between branches and paths.

28. A computer system for navigating a virtual environment, the computer system comprising:

- a computer;
- a display device connected to the computer;
- one or more input devices connected to the computer; and
- a computer program executing on the computer, wherein the computer program comprises computer instructions for:
  - displaying a map of at least a portion of a virtual environment generated from a real

environment in a first window of the display device;

displaying a visual representation of the virtual environment in a second window of the display device; and

5

highlighting a branch in the map displayed in the first window while a visual representation of a corresponding path in the real environment is displayed in the second window;

10

detecting that a user has selected an intersection on the map;

in response to the user selecting the intersection, displaying a corresponding visual representation of the intersection in the virtual environment in the second window;

15

highlighting an icon displayed in the first window and corresponding to an element of the virtual environment associated with multimedia data while the element is displayed in the second window;

20

detecting that a user has selected a visual representation of an element of the virtual environment associated with multimedia data; and

in response to the user selecting the element, displaying the multimedia data in a window of the display device.

25

29. The computer system of claim 28, wherein the computer program is a video game.

30

30. The computer system of claim 28, wherein the virtual environment further comprises a visual representation of an exhibit.

31. A method of navigating a virtual environment generated from a real environment, the method comprising:

5 displaying a map of at least a portion of the virtual environment in a first window of a display device of a computer;

displaying a visual representation of the virtual environment in a second window of the display device; and

10 highlighting a branch in the map displayed in the first window while a visual representation of a corresponding path in the real environment is displayed in the second window;

15 detecting that a user has selected an intersection on the map;

in response to the user selecting the intersection, displaying a corresponding visual representation of the intersection in the virtual environment in the second window;

20 highlighting an icon displayed in the first window and corresponding to an element of the virtual environment associated with multimedia data while the element is displayed in the second window;

25 detecting that a user has selected a visual representation of an element of the virtual environment associated with multimedia data; and

30 in response to the user selecting the element, displaying the multimedia data in a window of the display device.

32. A computer-readable storage medium comprising computer instructions for:

displaying a map of at least a portion of a

virtual environment generated from a real environment in a first window of a display device of a computer;

5 displaying a visual representation of the virtual environment in a second window of the display device; and

10 highlighting a branch in the map displayed in the first window while a visual representation of a corresponding path in the real environment is displayed in the second window;

detecting that a user has selected an intersection on the map;

15 in response to the user selecting the intersection, displaying a corresponding visual representation of the intersection in the virtual environment in the second window;

20 highlighting an icon displayed in the first window and corresponding to an element of the virtual environment associated with multimedia data while the element is displayed in the second window;

detecting that a user has selected a visual representation of an element of the virtual environment associated with multimedia data; and

25 in response to the user selecting the element, displaying the multimedia data in a window of the display device.

33. A computer system for navigating a virtual environment, the computer system comprising:

a server computer;

a global-area network;

at least one client computer connected to the server computer via the global-area network;

a display device connected to the client computer;  
one or more input devices connected to the client  
computer; and

a computer program executing on the client  
5 computer, wherein the computer program comprises  
computer instructions for:

retrieving a virtual environment generated  
from a real environment from a storage of the  
server computer;

10 displaying a map of at least a portion of the  
virtual environment in a first window of the  
display device;

displaying a visual representation of the  
virtual environment in a second window of the  
15 display device; and

enabling a user to navigate through the  
virtual environment by entering navigation  
commands via at least one input device.

20 34. A method of selling goods on-line, the method  
comprising:

generating a virtual store environment  
generated from a real store environment, the  
virtual environment comprising a plurality of  
25 nodes connected by branches, wherein each node  
represents an intersection in the real environment  
and each branch represents a path connecting  
intersections in the real environment;

displaying a map of at least a portion of the  
30 nodes and branches of the virtual store  
environment in a first window of a display device  
of a computer;

displaying a visual representation of the  
virtual store environment in a second window of



the display device;

a user navigating through the virtual store environment by entering navigation commands via at least one input device connected to the display device;

a user selecting for purchase at least one item displayed in the second window; and

completing a purchase transaction on-line to transfer ownership of the selected item to the user.

35. The method of claim 34, wherein the computer is connected to a server computer via a global-area network and the virtual store environment is stored on the server computer.

36. The method of claim 34, wherein the user can obtain further information about an item by selecting an element displayed in the first or the second window associated with the item.

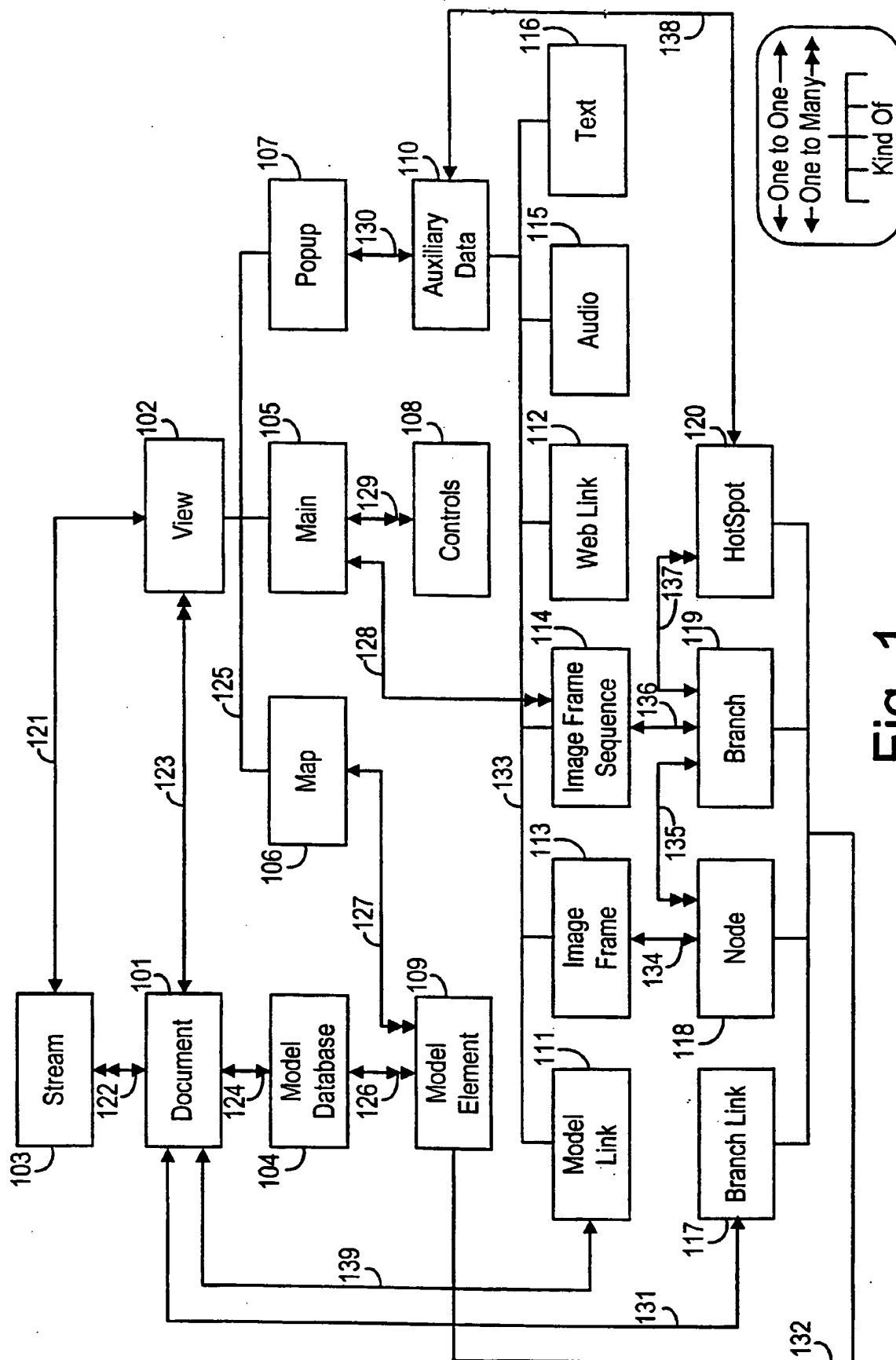


Fig. 1

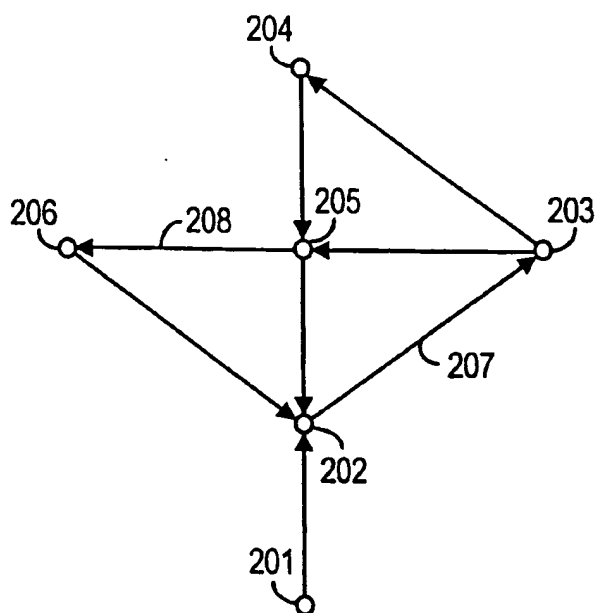


Fig. 2

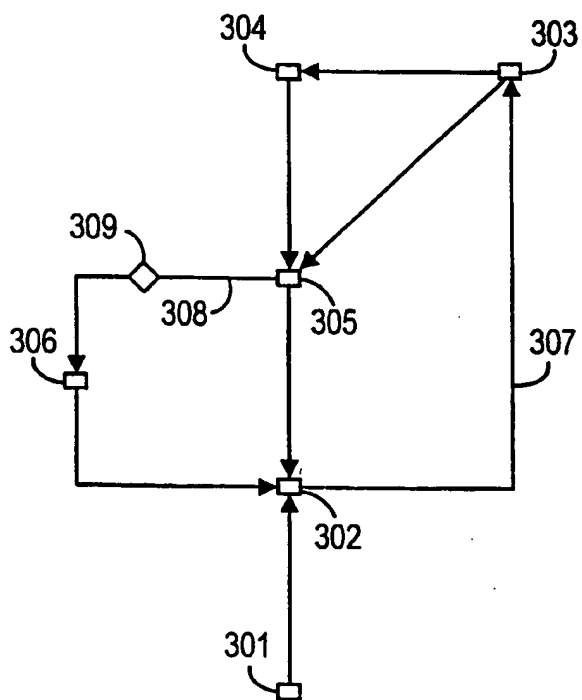
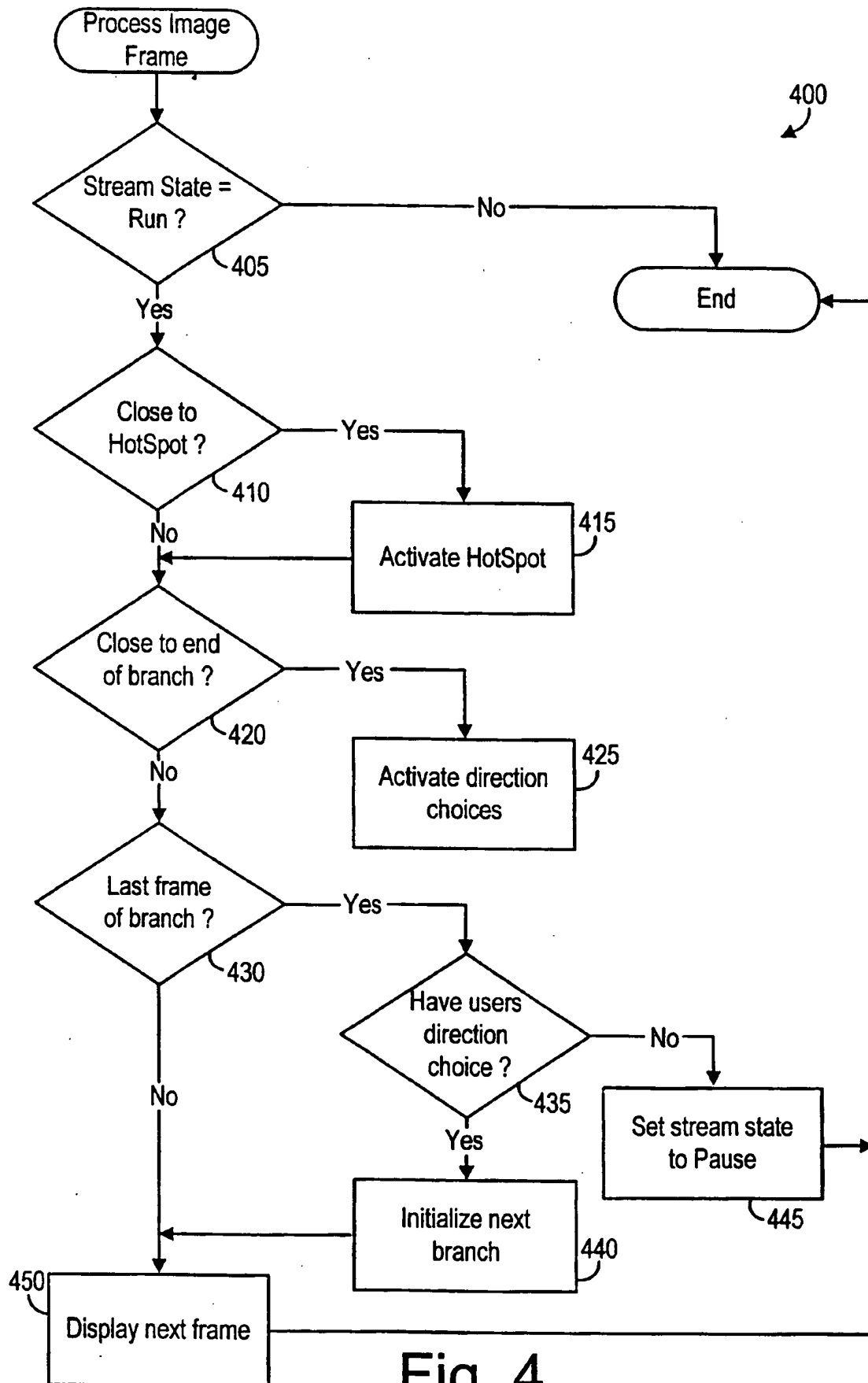


Fig. 3

**Fig. 4**

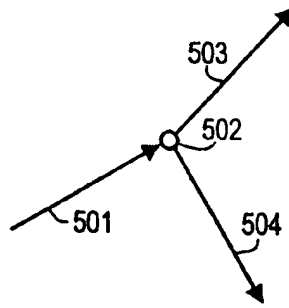


Fig. 5a

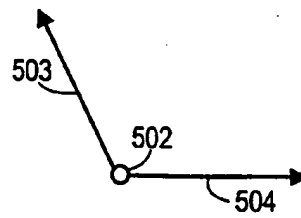


Fig. 5b

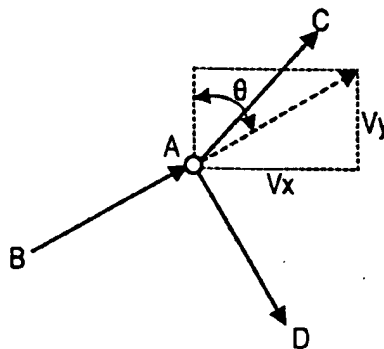
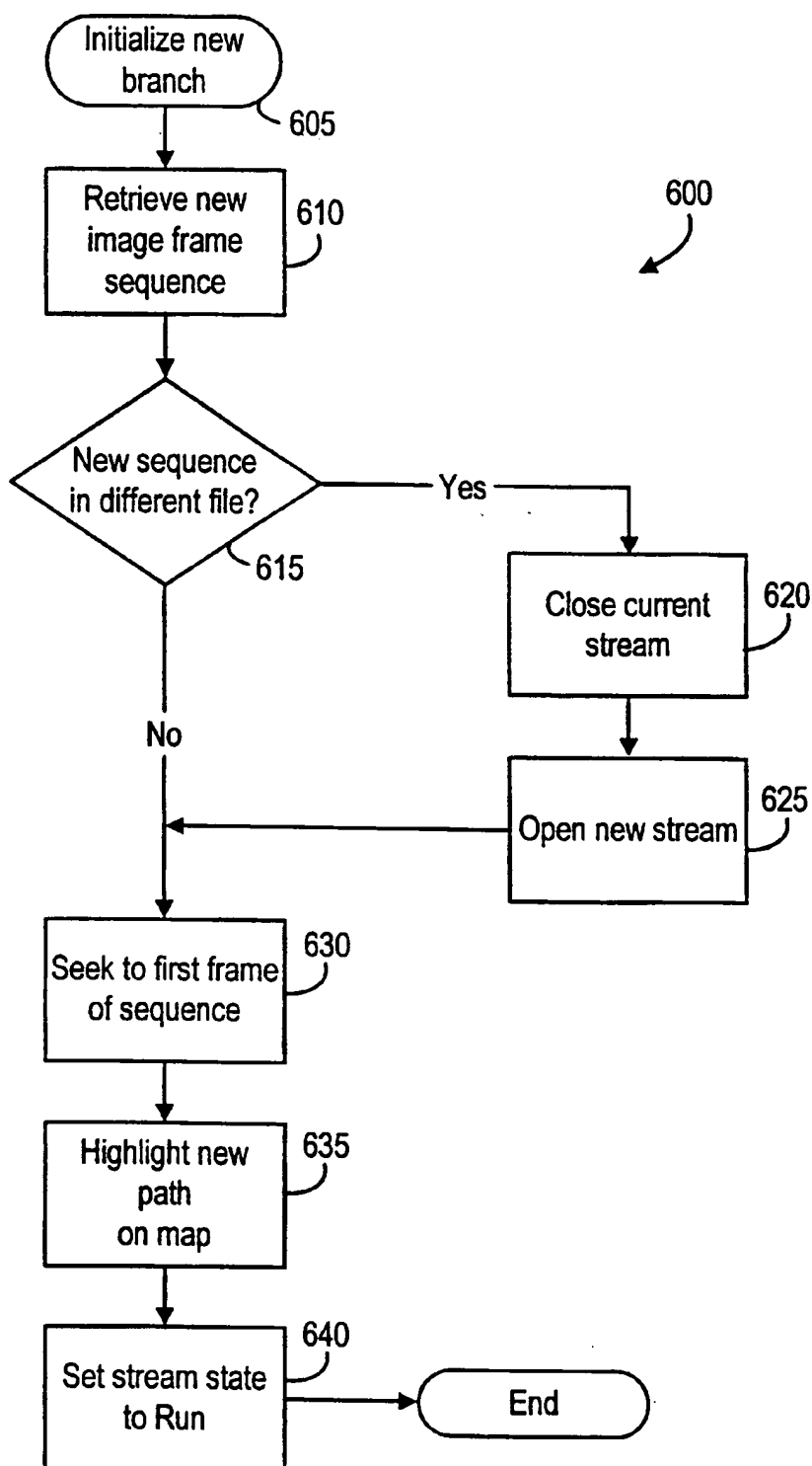


Fig. 5c

**Fig. 6**

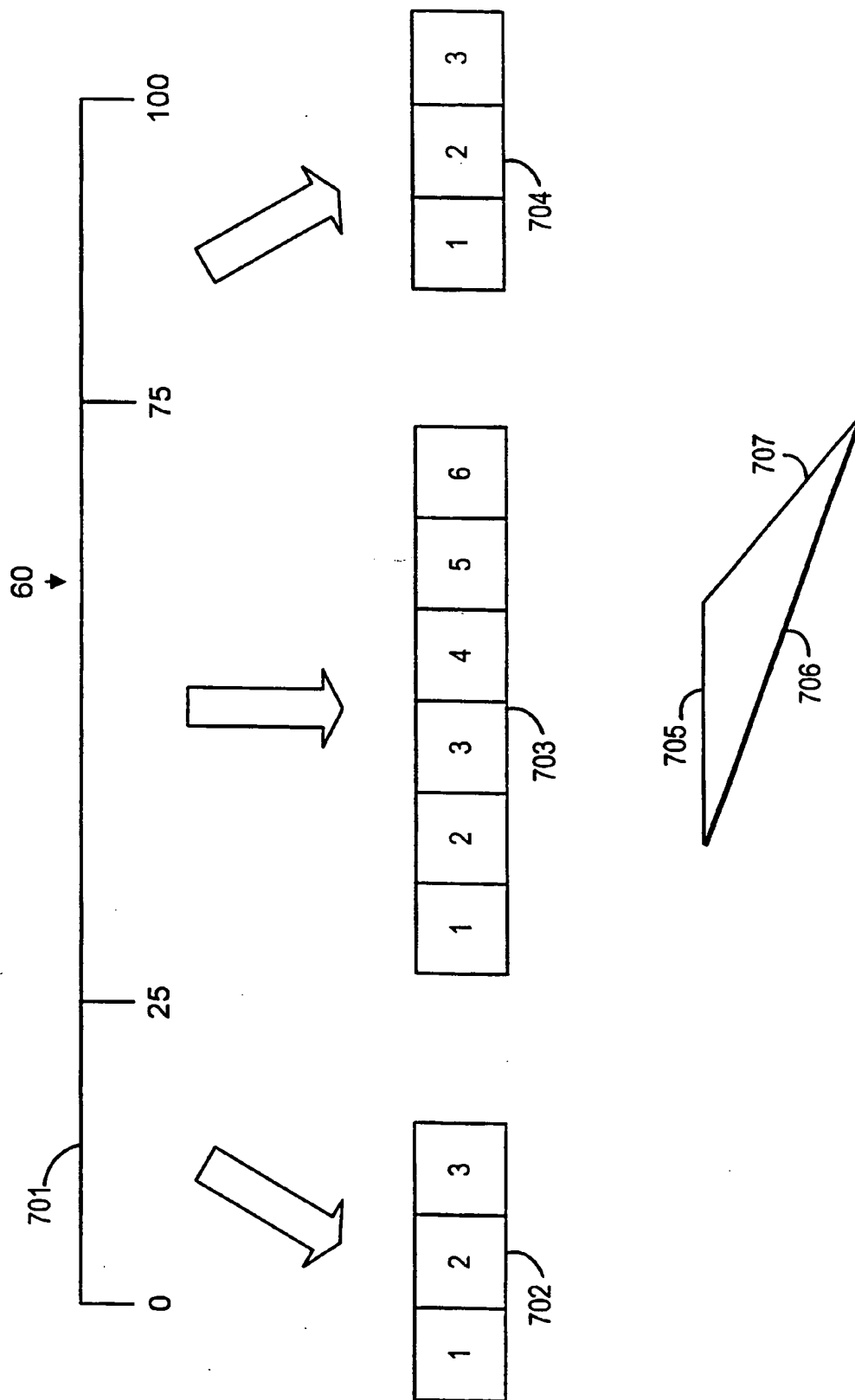


Fig. 7

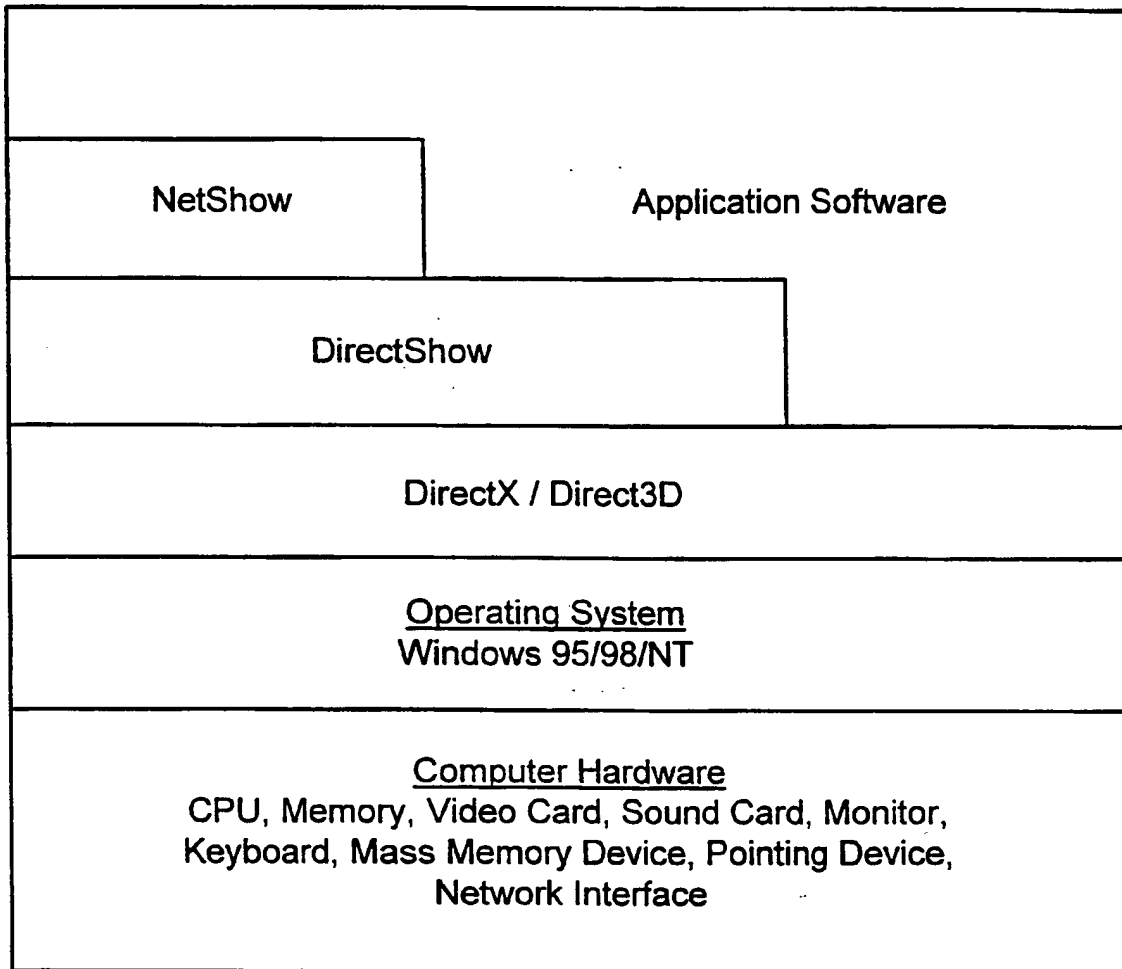


Fig. 8



Fig. 9

Fig. 9A
Fig. 9B

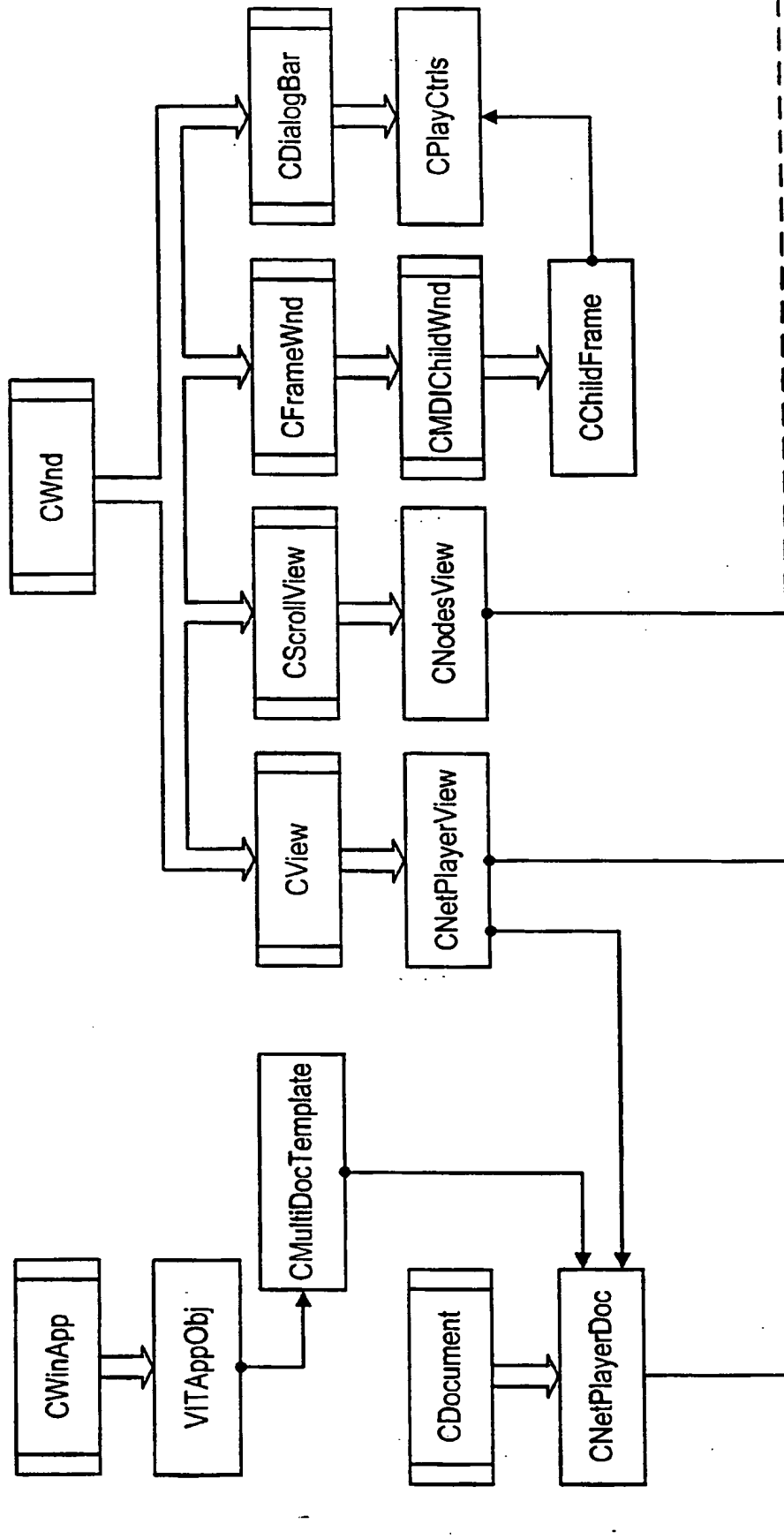
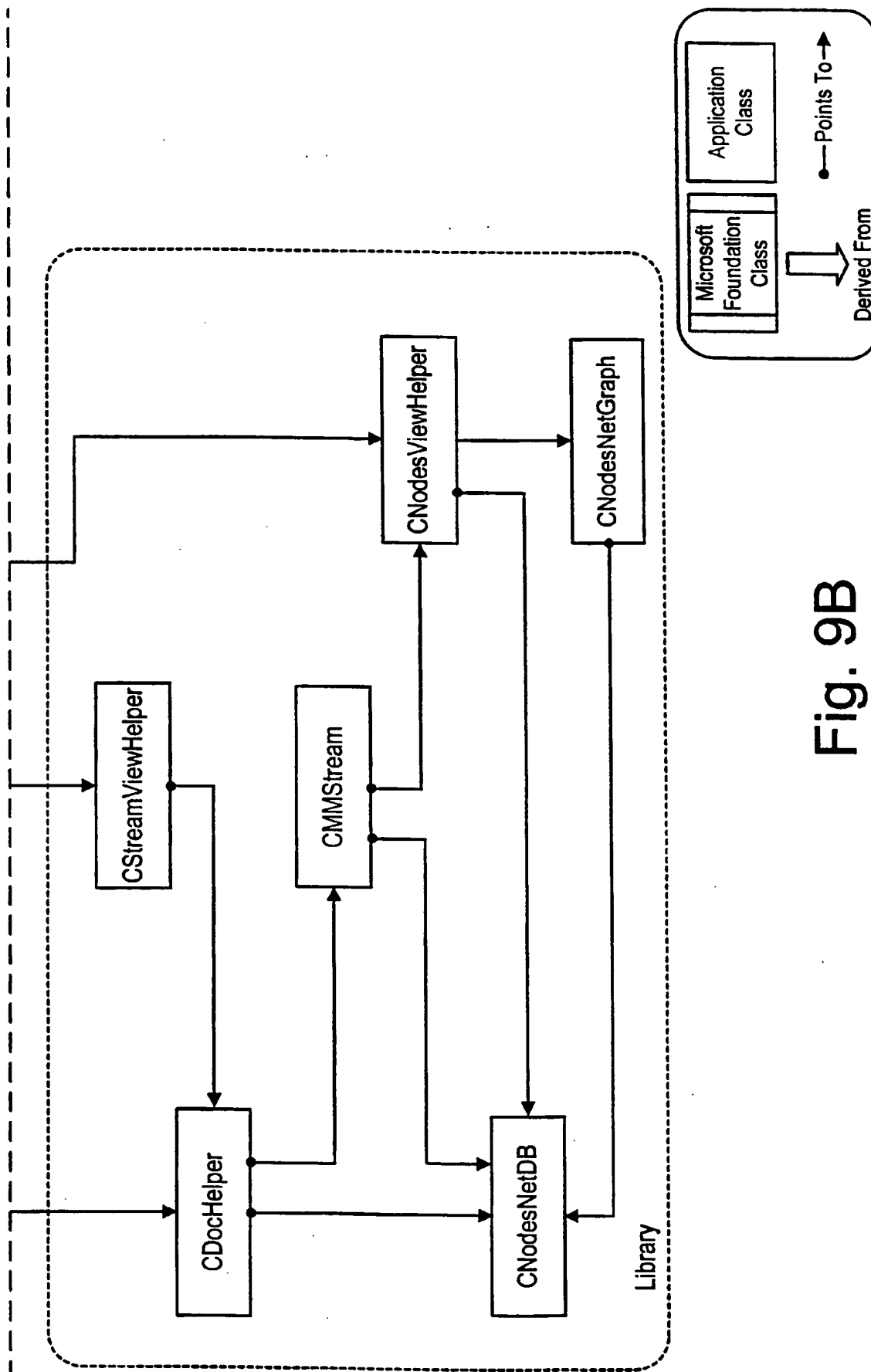


Fig. 9A



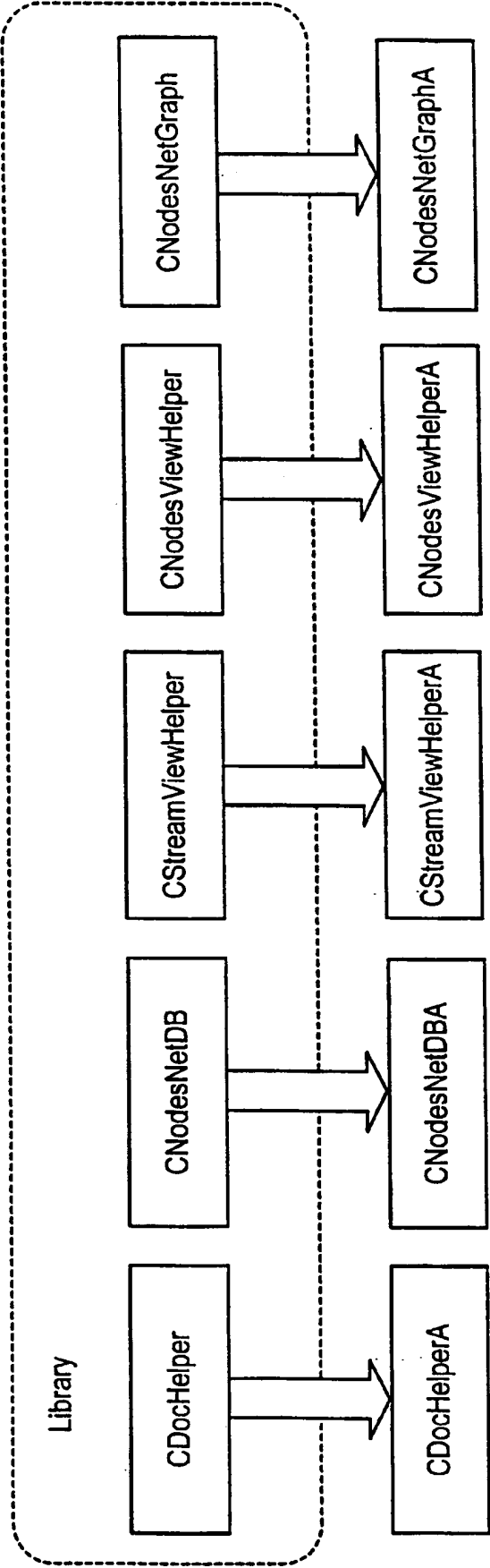
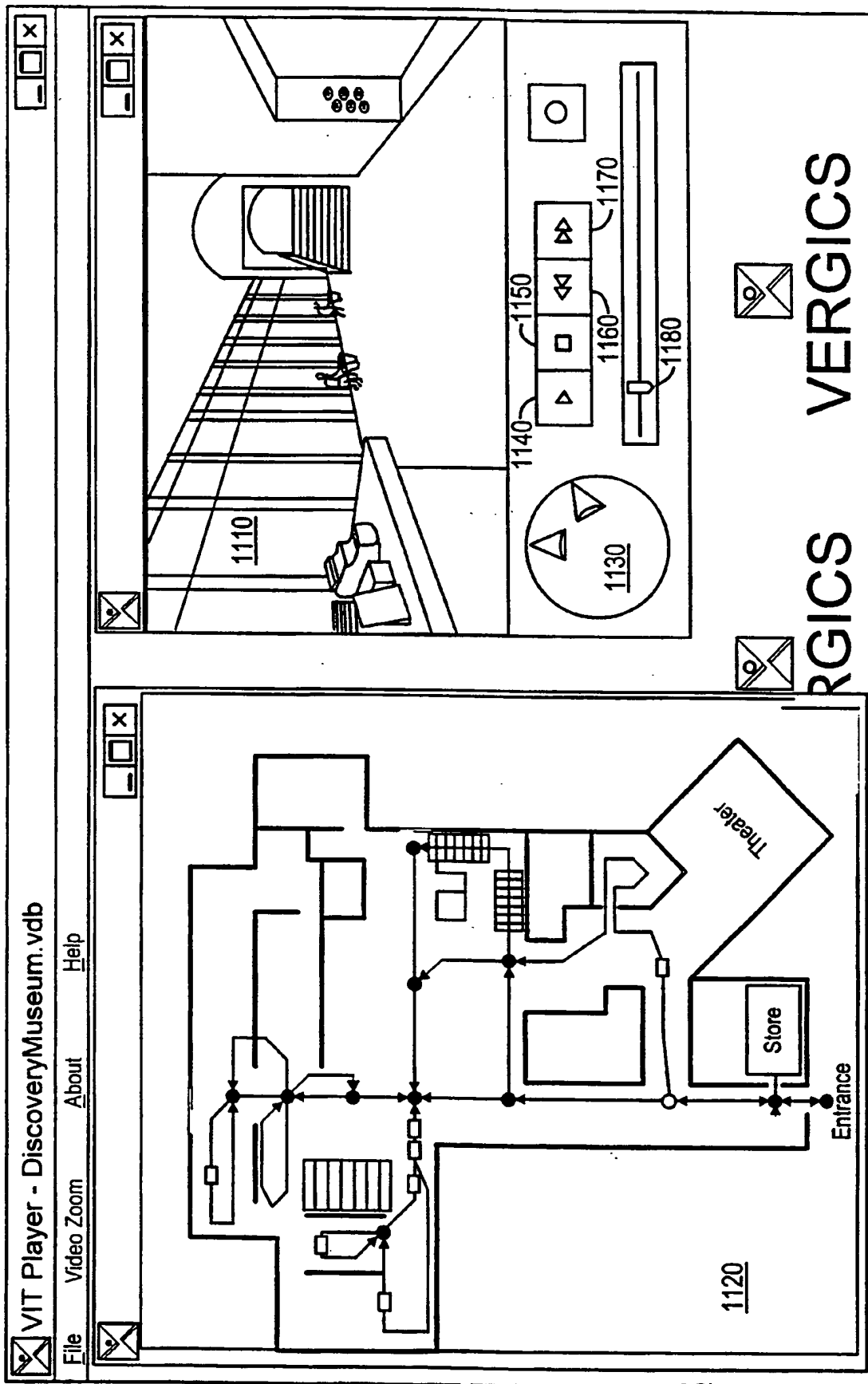


Fig. 10



1100

Fig. 11

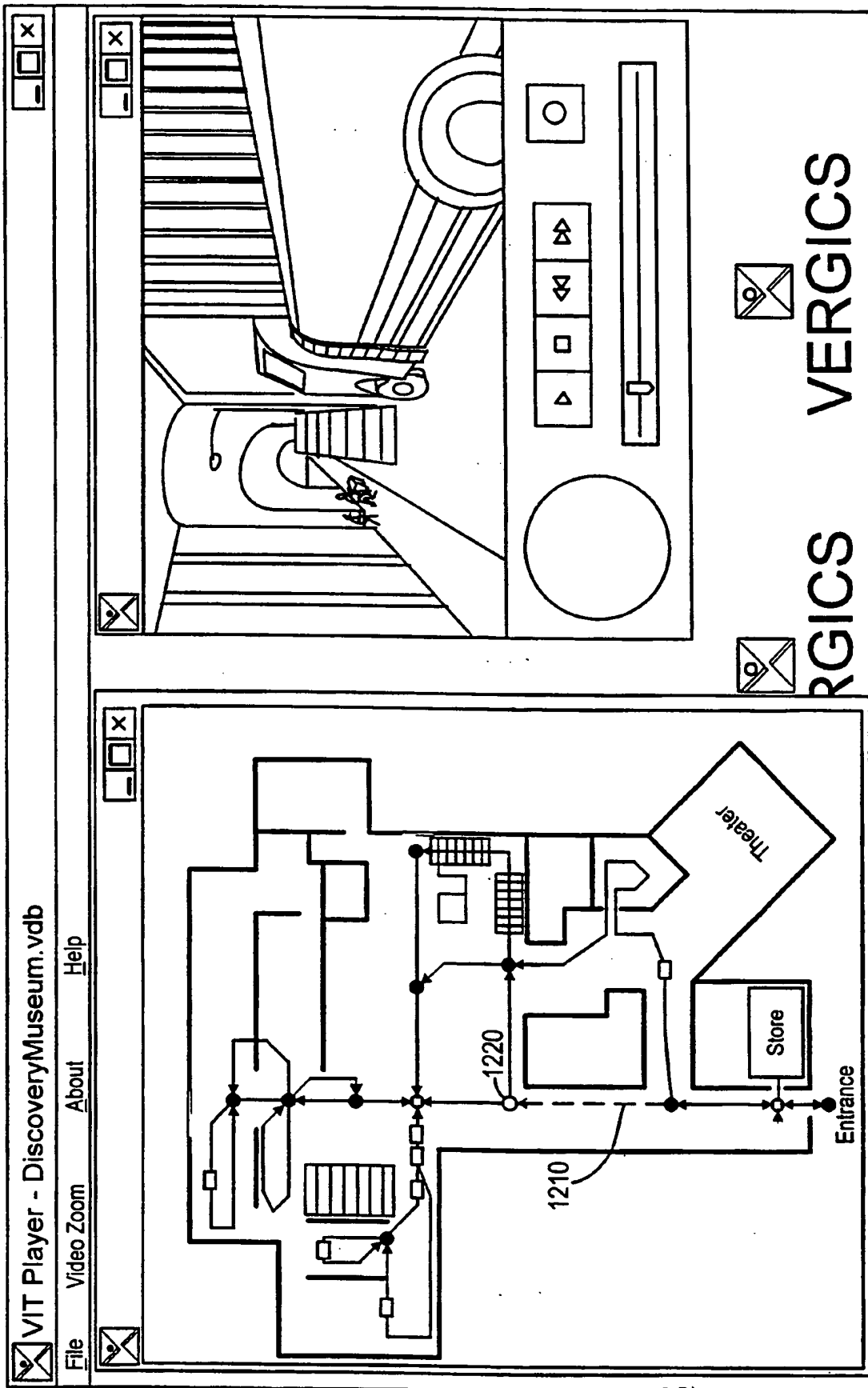


Fig. 12

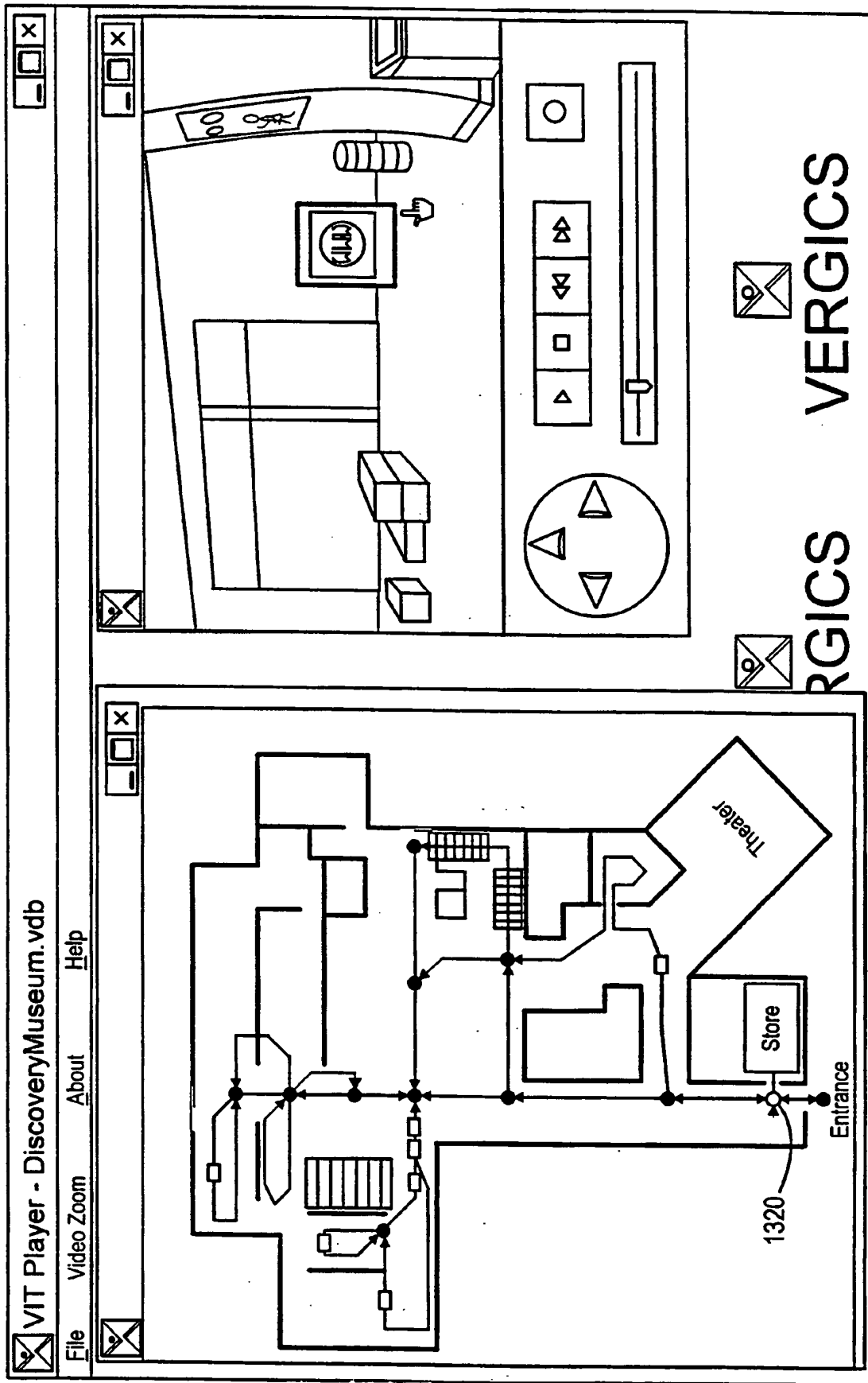


Fig. 13

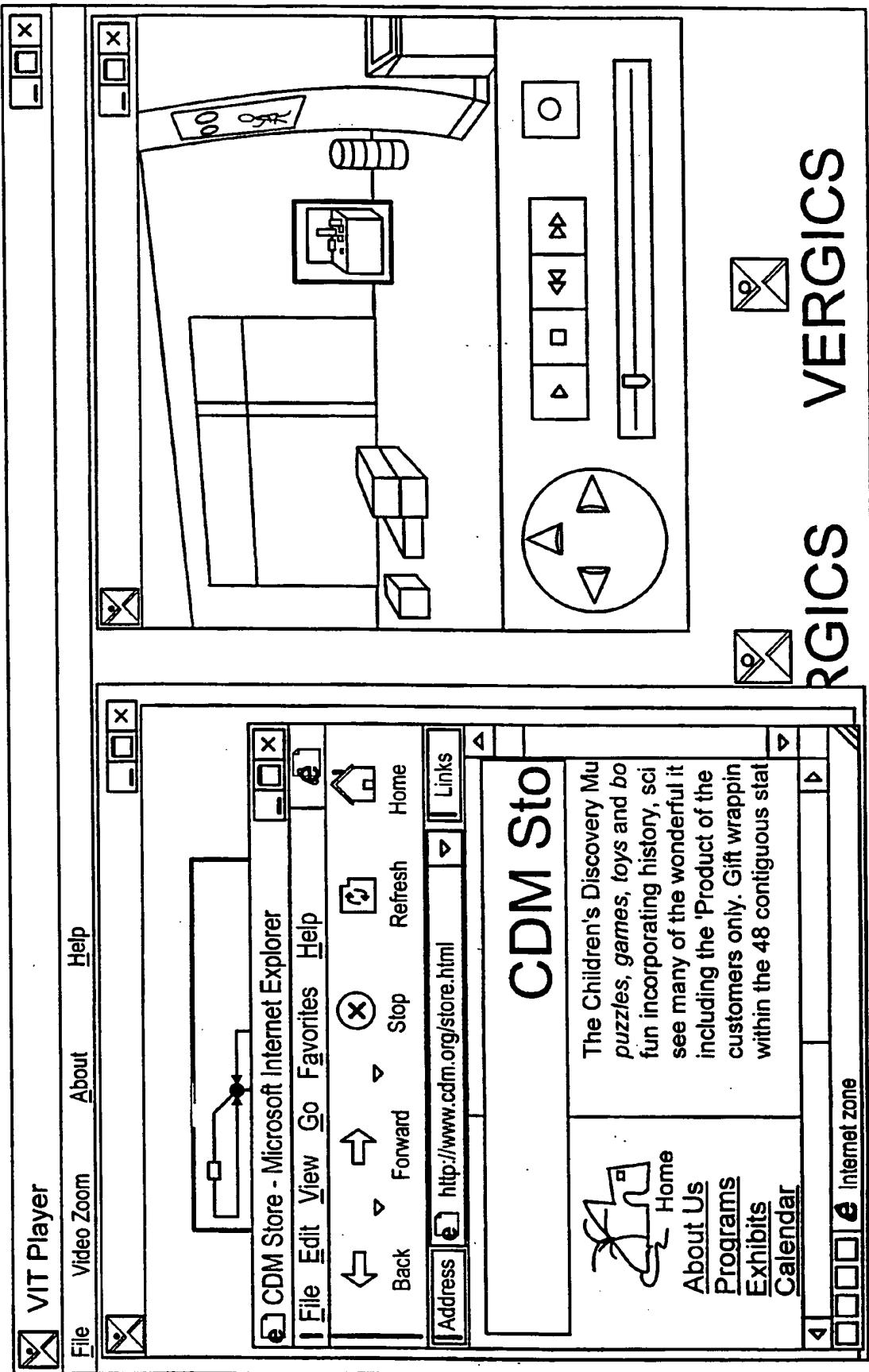


Fig. 14

1410

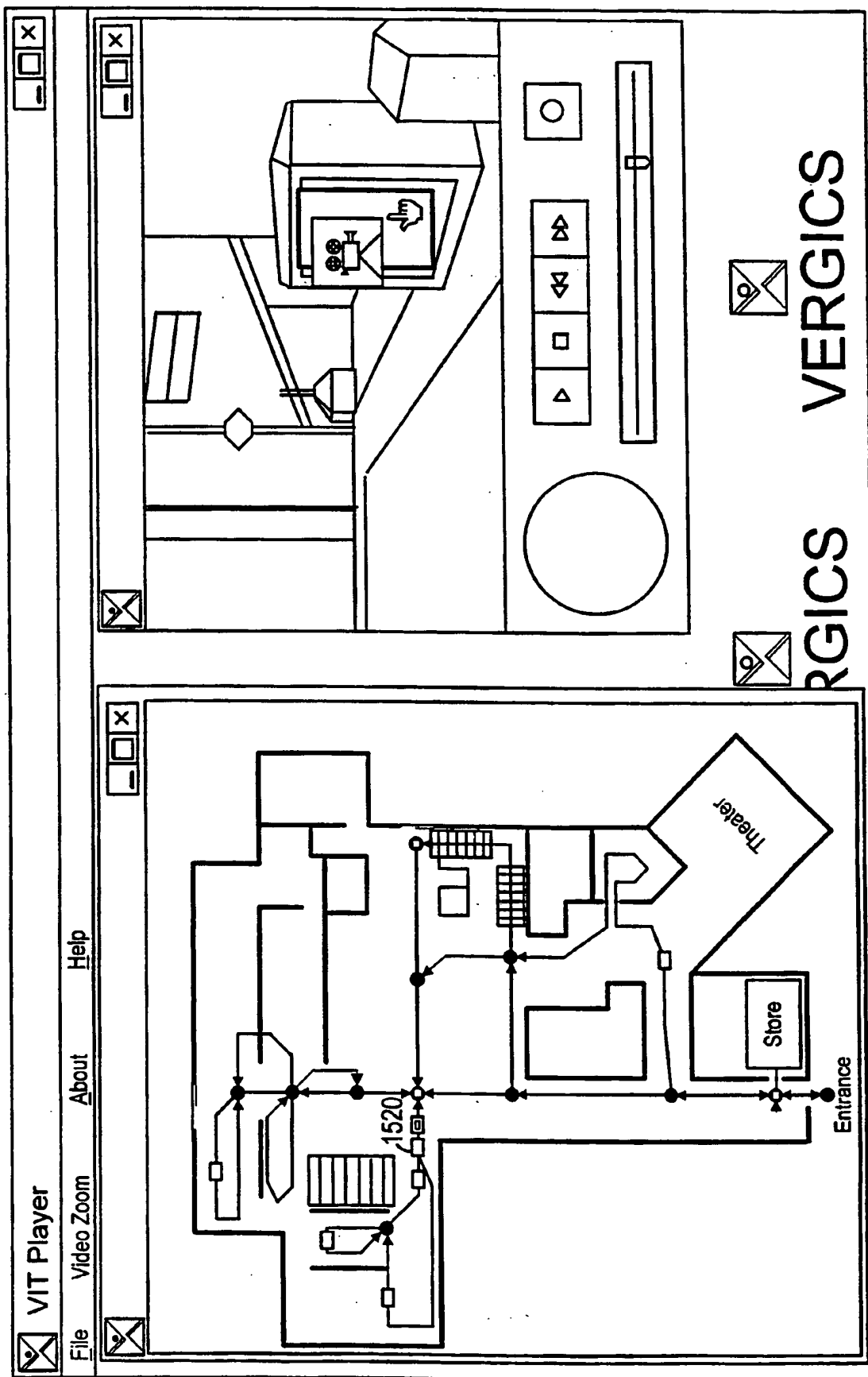


Fig. 15



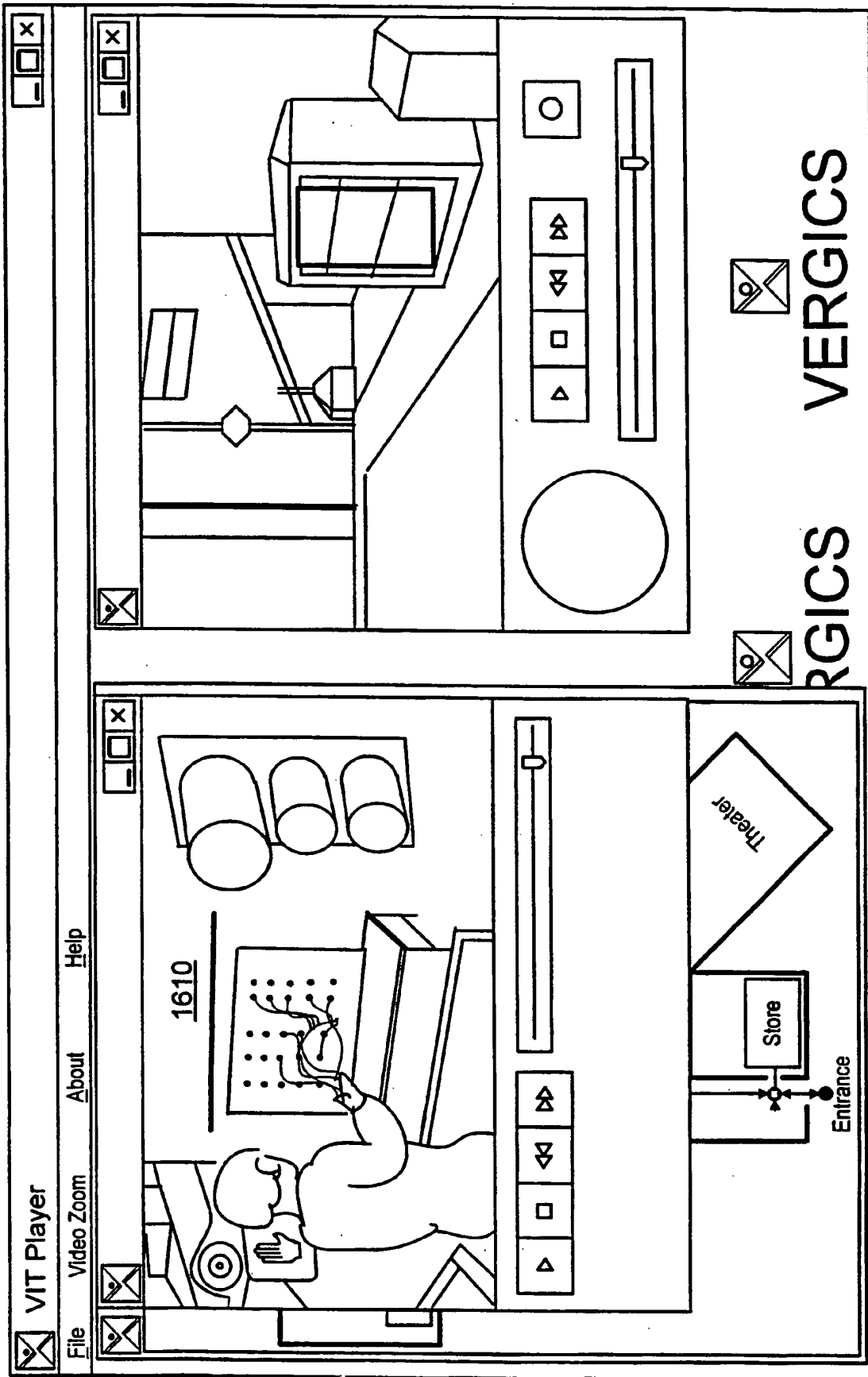


Fig. 16

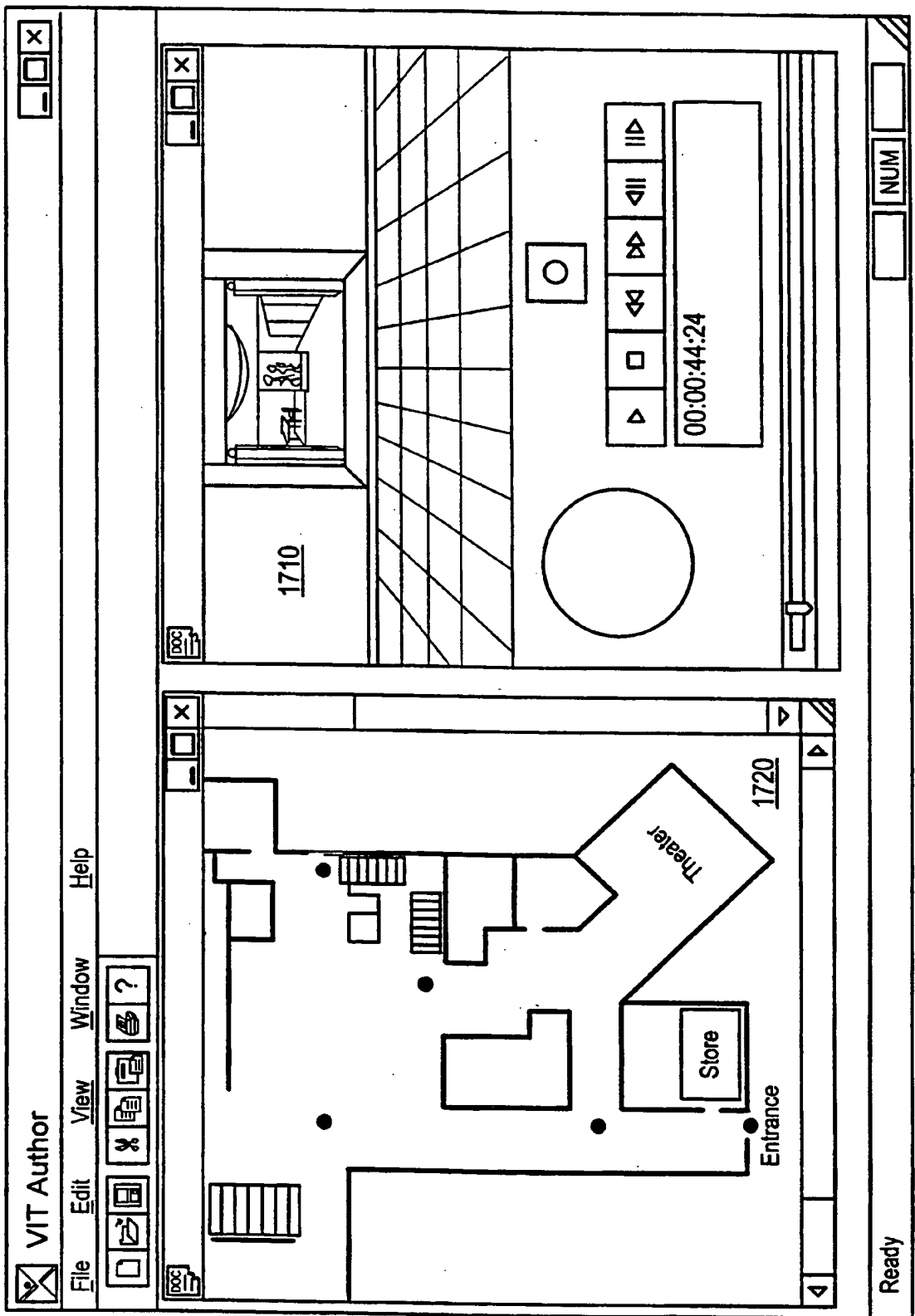


Fig. 17

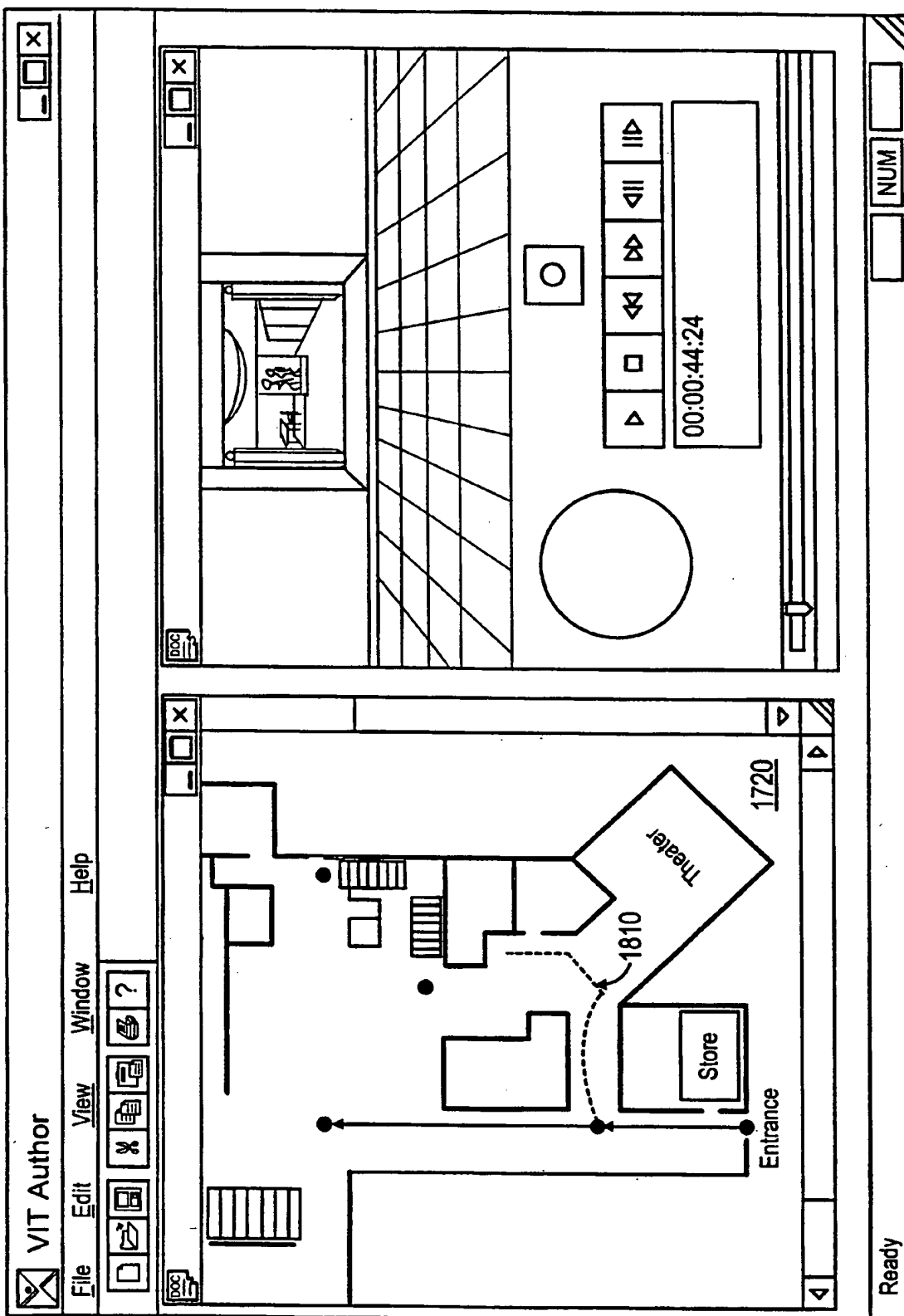


Fig. 18

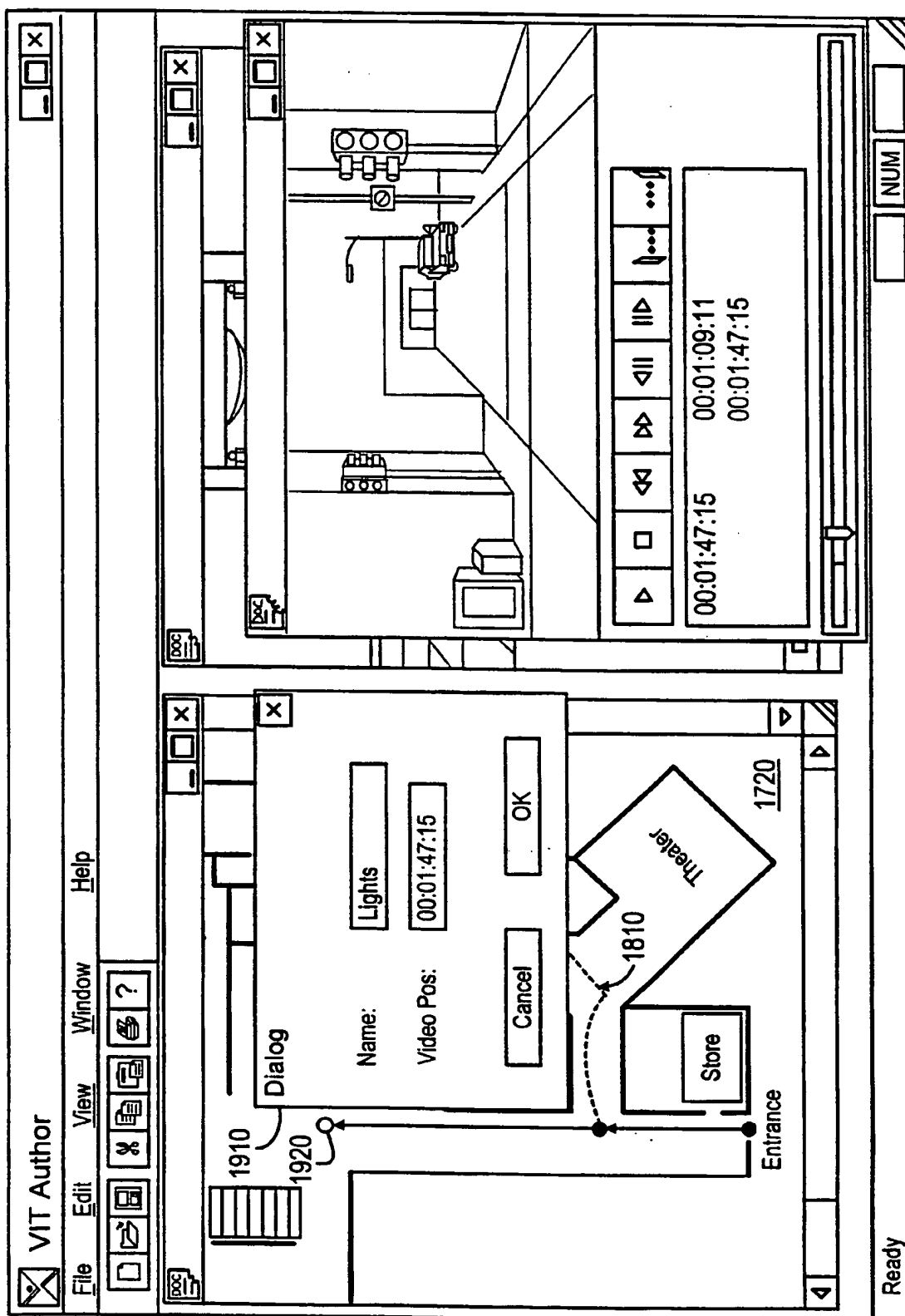


Fig. 19

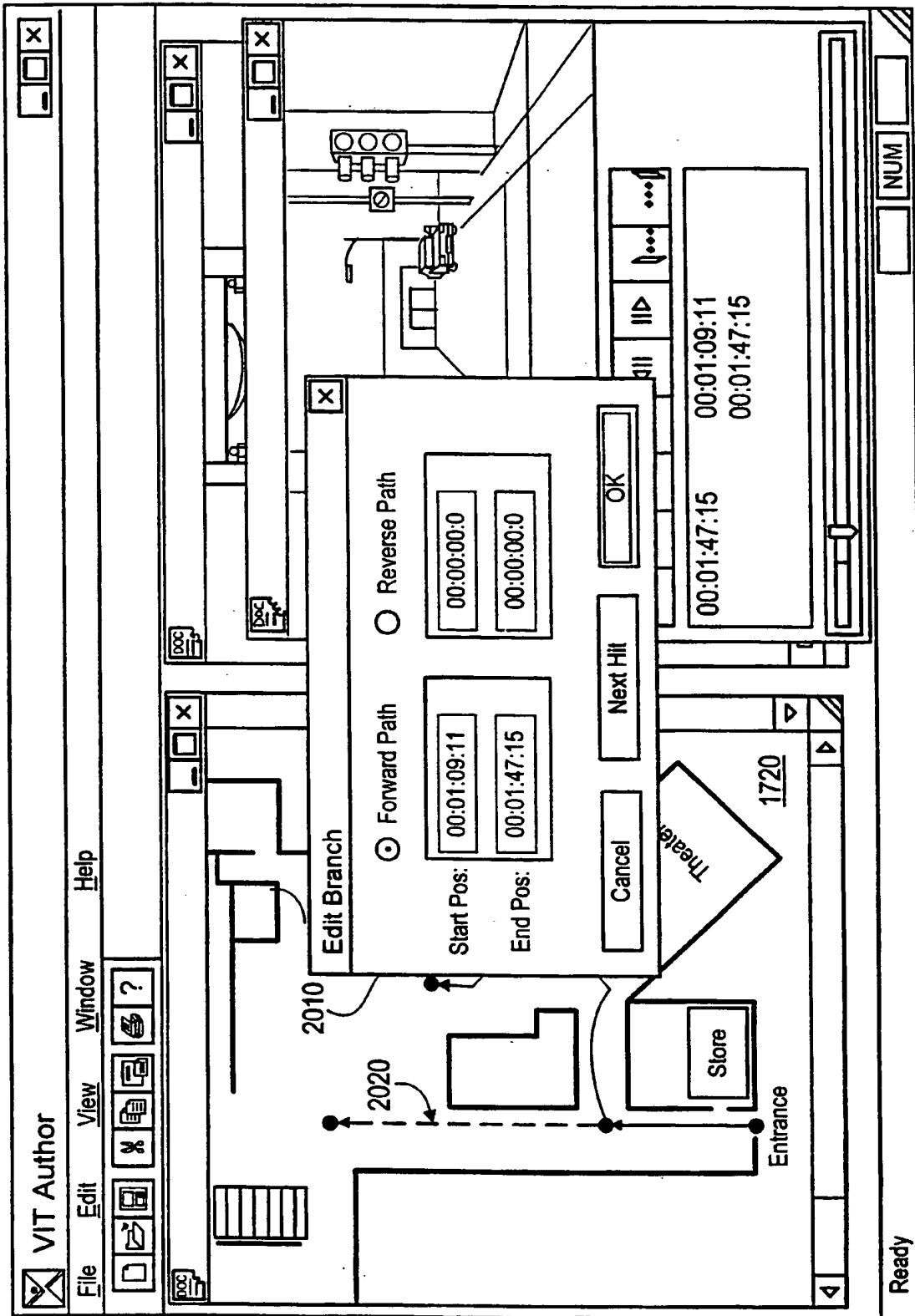


Fig. 20

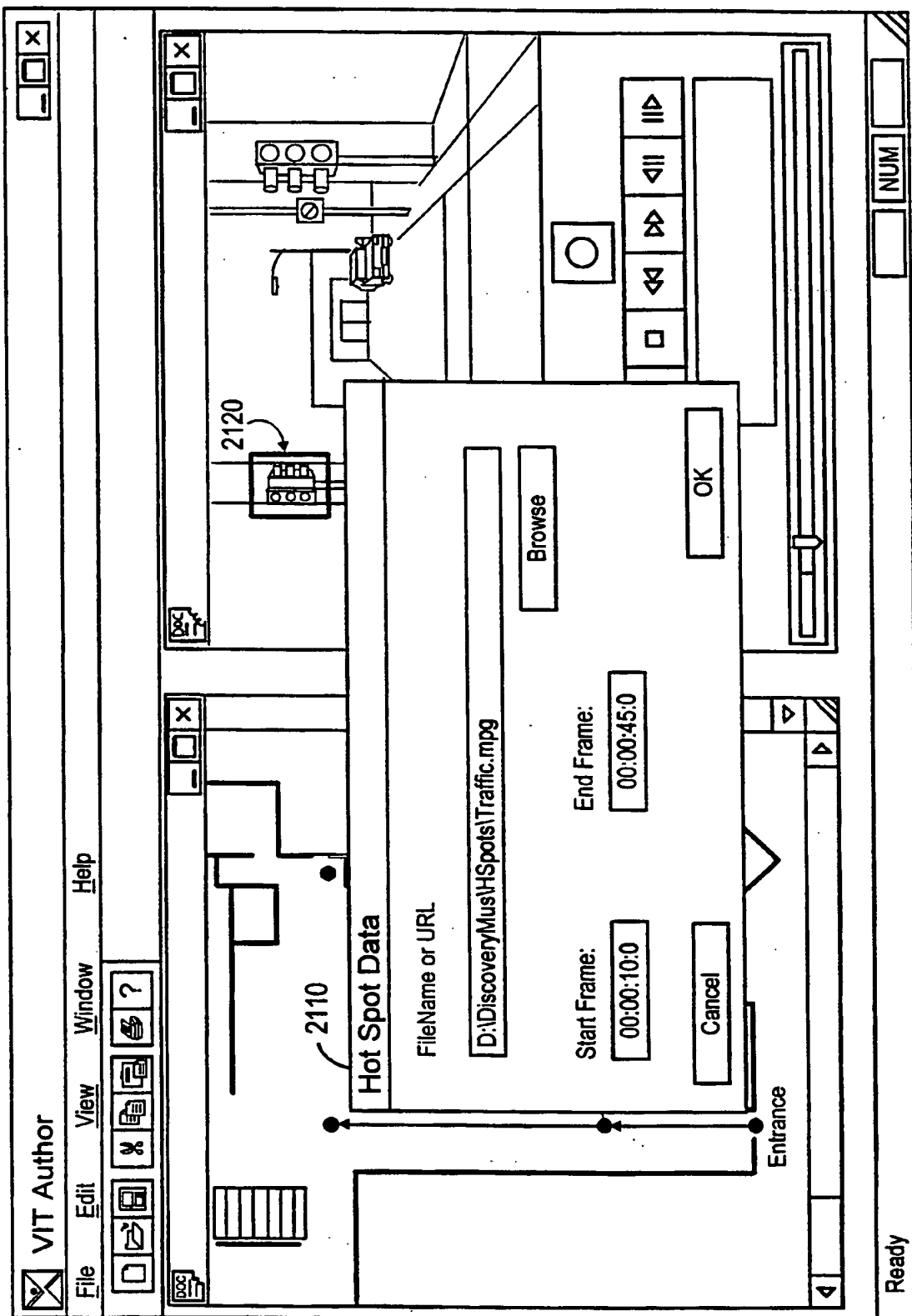


Fig. 21

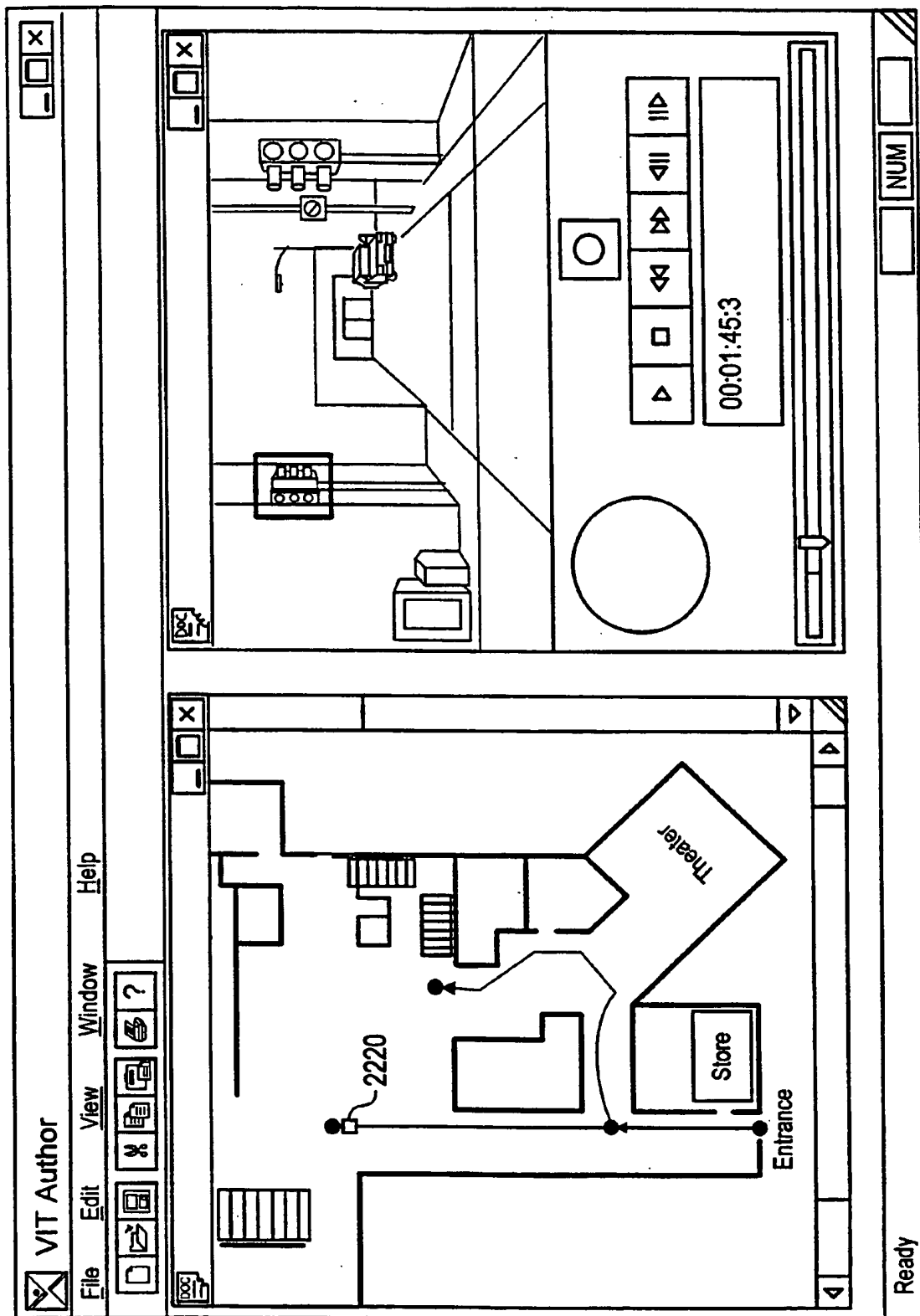


Fig. 22

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/07623

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F3/033

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 602 564 A (KAZUAKI IWAMURA ET AL) 11 February 1997 (1997-02-11)	1,3,4,7, 8,10,12, 13,16, 17,19, 21,22, 25,26
Y		5,6,14, 15,23,24
A	column 2, line 50 -column 3, line 60  column 5, line 52 -column 6, line 10 column 7, line 7 - line 65 column 10, line 62 -column 12, line 46 figures 2A-C,8,14B  --- -/--	2,9,11, 18,20, 27,28, 31,32,34



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

\* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"Z" document member of the same patent family

Date of the actual completion of the international search

17 July 2000

Date of mailing of the international search report

28/07/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Kröner, S



# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/07623

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 660 289 A (AISIN AW CO) 28 June 1995 (1995-06-28)	1,2,7, 10,11, 16,19, 20,25
A	column 1, line 5 -column 2, line 38  column 7, line 1 -column 8, line 56 column 11, line 38 -column 12, line 4 column 34, line 27 -column 35, line 5 figures 3,4A,6,30,31,48	3,9,12, 18,21,27
Y	PICH J: "3D IM WEB" CHIP ZEITSCHRIFT FUER MIKROCOMPUTER-TECHNIK,DE,VOGEL VERLAG. WURZBURG, no. 2, 1 February 1997 (1997-02-01), pages 192-195, XP000656986 ISSN: 0170-6632	5,6,14, 15,23,24
A	page 193, right-hand column, line 18 - line 35	33
A	DARKEN R P ET AL: "Map usage in virtual environments: orientation issues" PROCEEDINGS IEEE VIRTUAL REALITY (CAT. NO. 99CB36316), PROCEEDINGS OF VIRTUAL REALITY, HOUSTON, TX, USA, 13-17 MARCH 1999, pages 133-140, XP002142329 1999, Los Alamitos, CA, USA, IEEE Comput. Soc, USA ISBN: 0-7695-0093-5 section 6.4 figures 3,4	28,29, 31,32

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/07623

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5602564 A	11-02-1997	JP 5134600 A	28-05-1993
		JP 5135154 A	01-06-1993
		CA 2082897 A	15-05-1993
		US 5945976 A	31-08-1999
EP 0660289 A	28-06-1995	JP 2824198 B	11-11-1998
		JP 7190790 A	28-07-1995
		JP 7198401 A	01-08-1995
		JP 2675970 B	12-11-1997
		JP 7234993 A	05-09-1995
		US 5612881 A	18-03-1997
		US 5787383 A	28-07-1998